KI ABSICHERUNG
Safe AI for Automated Driving

KI Absicherung - Finale Ergebnissteckbriefe TP1

Version zu Veröffentlichung

Version	1.0
Editor	Fridolin Bauer / BMW Group Dr. Stephan Scholz / Volkswagen AG PD Dr. Michael Mock / Fraunhofer IAIS
Projektkoordination	Volkswagen AG / Fraunhofer IAIS
Fälligkeit	31.12.2022
Erstellungsdatum	02.11.2022

Gefördert durch:

Bundesministerium
für Wirtschaft
und Klimaschutz

aufgrund eines Beschlusses
des Deutschen Bundestages

Dokumenteninformation

Autoren

Andreas Blattmann / Unversitäti-Heidelberg

Nikolas Brasch / Technische Universität München

Patrick Feifel / Stellantis, Opel Automobile GmbH

Juncong Fei / Stellantis, Opel Automobile GmbH

Michael Fuerst / DFKI

Nicolas Gay / DXC Luxoft

Stefano Gasperini / BMW Group

Oliver Grau / Intel Corporation

Korbinian Hagn / Intel Corporation

Philipp Heidenreich / Stellantis, Opel Automobile GmbH

Simon Heming / Robert Bosch GmbH

Falk Heuer / ZF Friedrichshafen AG

Falk Kappel / ZF Friedrichshafen AG

Anja Kleinke / Valeo Schalter und Sensoren GmbH

Timo Saemann / Valeo Schalter und Sensoren GmbH

Mahdi Saleh / Technische Universität München

Loren Schwarz / BMW Group

Bharat Shinde / Valeo Schalter und Sensoren GmbH

Thomas Stone / BMW Group

Alexander Woelker / ZF Friedrichshafen AG

Corinna Wright / Robert Bosch GmbH

Reviewer

Fridolin Bauer / BMW Group

Stephan Scholz / Volkswagen AG

Michael Mock / Fraunhofer IAIS

Andreas Blattmann / Universität-Heidelberg

Nikolas Brasch / Technische Universität München

Michael Fuerst / DFKI

Nicolas Gay / DXC Luxoft

Korbinian Hagn / Intel Corporation

Philipp Heidenreich / Stellantis, Opel Automobile GmbH

Simon Heming / Robert Bosch GmbH

Barat Shinde / Valeo Schalter und Sensoren GmbH

Kontakt

(in Vertretung für die Projektkoordination)

European Center for Information and Communication Technologies – EICT GmbH

EUREF-Campus Haus 13

Torgauer Straße 12-15

10829 Berlin

Germany

Email: projects@eict.de

Projektwebsite: https://www.ki-absicherung-projekt.de/

Revisionslog

Version	Datum	Kommentar	Autor	Partner
0.1	Bis 21.10.2022	Input auf Confluence	s.o.	s.o.
0.2	21.-24.10.2022	Ausspielen Word, Prüfung und Korrektur Übertrag, Strukturierung und Layout Dokument	Bert Hildebrandt	EICT
0.3	24.10.2022	Formelles Review und Layout	Dr. Nikos Papamichail	EICT
1.0	02.11.2022	Finalisierung	Dr. Nikos Papamichail	EICT

Inhaltsverzeichnis

1 AP1.1 - Technische Plattform

1.1 E1.1.1 Final: nur projektintern für KI Absicherung verfügbar

1.2 E1.1.2a Final: nur projektintern für KI Absicherung verfügbar

1.3 E1.1.2b Final: nur projektintern für KI Absicherung verfügbar

1.4 E1.1.2c Final: nur projektintern für KI Absicherung verfügbar

1.5 E1.1.2d Final: nur projektintern für KI Absicherung verfügbar

1.6 E1.1.3a Final: nur projektintern für KI Absicherung verfügbar

1.7 E1.1.3b Final: nur projektintern für KI Absicherung verfügbar

1.8 E1.1.3c Final: nur projektintern für KI Absicherung verfügbar

1.9 E1.1.3d Final: nur projektintern für KI Absicherung verfügbar

2 AP1.2 - Anforderungen an die KI-Funktion

2.1 E1.2.1 Final: nur projektintern für KI Absicherung verfügbar

2.2 E1.2.2 Final: nur projektintern für KI Absicherung verfügbar

2.3 E1.2.3 Final: nur projektintern für KI Absicherung verfügbar

2.4 E1.2.4 Final: Definition von nominalen Basisszenarien (zur Veröffentlichung)

2.4.1 Formal Classification

Criteria	Classification according to VHB
Type of result	*Document*
Group/Cluster	
Type of content	*Definition*
Classification level	*PU*

Authors: Loren Schwarz, Philipp Heidenreich, Frederik Blank, Simon Heming, Thomas Schulik

2.4.2 Description of the result

This result aims at describing the properties that the synthetic dataset shall have, so that a deep learning based pedestrian detection algorithm can be trained and tested successfully. To this end, a set of requirements is developed that specifies the base scenarios of the dataset, which includes the data variation and the amount of sequences and single frames with low correlation.

During the project a strong collaboration between similar results, e.g. E2.2.6, and E4.1.1, has been enabled by an active participation in P1.

A short overview of the used terminology is listed below.

- Base context: configuration of static context elements and definition of possible dynamic context elements
- Scenario: inspired from PEGASUS project, different abstraction levels are functional, logical, and concrete scenario
- Scene storyboard: complete specification of the input for the data generation toolchain, all dimensions of the input space are specified, the input space corresponds to the operational design domain described by Zwicky boxes

The focus of E1.2.4/ E1.2.5 shall be the direct influence on the dataset used to train and test the pedestrian detection methods, in terms of functional and logical scenarios.

2.4.2.1 Similar public datasets

The generated synthetic dataset shall be similar in character to datasets, commonly used to train and test deep learning based pedestrian detection methods, such as the NuScenes dataset [1], the KITTI object detection dataset [2], or the AEV driving dataset [3]. An overview of the mentioned datasets is given in the table below

Dataset	Sensors	Annotation	Number of sequences	Number of keyframes	Number of objects
NuScenes	6 cameras 1 LiDAR 5 radar	3D bounding boxes	1000	40k	1400k
KITTI object detection dataset	1 stereo camera 1 LiDAR	2D/ 3D bounding boxes		15k	80k
AEV driving dataset	6 cameras 5 LiDAR	3D bounding boxes (1), Semantic pixel labeling (2)		12k (1), 40k (2)	

The NuScenes dataset consists of 1000 sequences, which contain interesting scenes with different traffic situations, mostly compliant with traffic rules, but also explicitely non-compliant with traffic rules. The sequences mostly contain scenes during daytime with good weather condition (sunny and cloudy), but also around 8% during night and 6% with rain. When compiling the sequences, it has been taken care of that the appearance is different so that the variation of the dataset is high.

The KITTI object detection dataset has been compiled in a similar way, however, single frames with a sufficiently different situation and appearance have been extracted from the recorded sequences. To demonstrate the high data variation in typical urban scenarios, four examples from the KITTI object detection dataset are shown below.

2.4.3 Approach

2.4.3.1 Generation of sequences and single frames

The base dataset consists of N_S sequences of consecutive frames with an average duration T_S and sampled at FPS frames per second. The number of sequences may result from a smaller number of distinct scenarios, that have been modeled using the data variation given below. Furthermore, there are N_E camera starting positions, where the camera is mounted on the ego vehicle and the ego vehicle drives on a trajectory according to typical traffic situations. The resulting total number of generated frames then is N_S*T_S*FPS*N_E. Assuming that only single frames with sufficiently low correlation, e.g. separated by 1 s, are useful to train and test a deep learning based pedestrian detection method, the number of useful single frames then is N_S*T_S*N_E. An overview of the mentioned quantities and a numerical example is given in the table below.

Quantity	Symbol	Dependency	Numerical example
Number of sequences	*N_S*		500
Average sequence duration	*T_S*		5 s
Frames per second	*FPS*		10 Hz
Number of starting positions	*N_E*		20
Number of frames		*N_S*T_S*FPS*N_E*	500.000
Number of frames with sufficiently low correlation		*N_S*T_S*N_E*	50.000

In AP1.3, the developed methods that are based on single frames are 2D bounding box detection (E1.3.3a) and semantic segmentation (E1.3.3b). Both methods have been predominantly investigated in TP3, whereas the remaining methods (E1.3.3c-E1.3.3e) that are based on sequences have been of little interest in TP3. Hence, a higher priority shall be given to the generation of single frames with sufficiently low correlation.

2.4.3.2 Data variation

A result from the P1 variation matrix vote [3] was that the most relevant domains/ parameters to vary are the static base context, the pedestrian properties (including relative position to interacting objects), the objects' surface and texture, the illumination, and the situation dynamics. We follow a similar structure in the following. Based on this first analysis, an extensive framework for the description of the operational design domain has been created using a Zwicky box approach [7].

Static base context

The agreed base context is a direct result of E4.1.1. Here we present some general statements about properties of a typical urban environment, as e.g. found in similar public datasets.

- Road infrastructure: The road infrastructure shall be according to a typical urban environment, where an initial setup may be a junction with 2x1 lanes including parts with adjacent sidewalks, parts with parallel parking spots or parking lanes, and parts with a dense building structure.
- Buildings: Buildings shall be of different type, e.g. single-family house with two floors, apartment building with three floors, office or shop buildings. Buildings shall have a different surface and texture, e.g. different windows/ doors or different facades. The building placement and separation shall be random and according to a typical urban appearance, e.g. 4-8 areas occupied and 1-2 areas free with vegatation.
- Vegetation: The vegetation shall consist of different trees, bushes and gras areas. The vegetation placement shall be random and according to a typical urban appearance.
- Static objects: Static objects represent candidates for false positive detections. Static objects should include parked cars, where the parking spots or parking lane results from the road

infrastructure. Regular parking spaces shall be occupied with 50%-80%. Static objects should also include advertising pillars, billboards, or parked bicycles.

Dynamic objects

- Pedestrians: Pedestrians are the primary object of interest, and therefore should have a large variability in appearance. The placement of pedestrians in the scene shall be uniformly balanced and according to a typical urban appearance, e.g. walking along the sidewalk or crossing the street. Some pedestrians shall be partly occluded, e.g. between parked cars. Occasionally, pedestrian groups shall be formed, e.g. at traffic lights or pedestrian crossings. Likewise, the poses of pedestrian assets shall be varied, they shall not be static but different in subsequent frames.
- Vehicles: Dynamic vehicles can be used to create different traffic situations. Moreover, vehicle placements can be used to hide certain areas of the scene. Vehicle may drive straight on the same lane in front of the ego vehicle or on the oncoming lane, or may take a turn.

Traffic situation with pedestrian

The traffic situation describes the behaviour of dynamic objects such as pedestrians and vehicles in the scenario. It shall contain mostly uncritical behaviour, but also critical behaviour that results in a collision, where the sequence ends before the collision. The traffic situation with critical behaviour may be based on the Euro NCAP test protocol for AEB VRU systems [4]. A corresponding hierarchy is sketched below.

- Car-to-pedestrian collision:
  - Straight road:
    - Pedestrian crossing farside: CPFA-50
    - Pedestrian crossing nearside:
      - No occlusion: CPNA-25, CPNA-75
      - Occluding vehicle: CPNC-50
      - Night condition: CPNA-50
    - Pedestrian walking longitudinal
  - Curved road/ junction
    - Turning right: CPTA-50
    - Turning left ...
- Non-critical:
  - Straight road ...
  - Curved road/ junction ...

Weather conditions

To specify weather conditions the following approximate occurrences are possible: dry and clear (65%), wet street (20%), rain (5%), fog with low visibility (5%), etc. Note that during the project, no realistic rain or fog models were available.

Light conditions

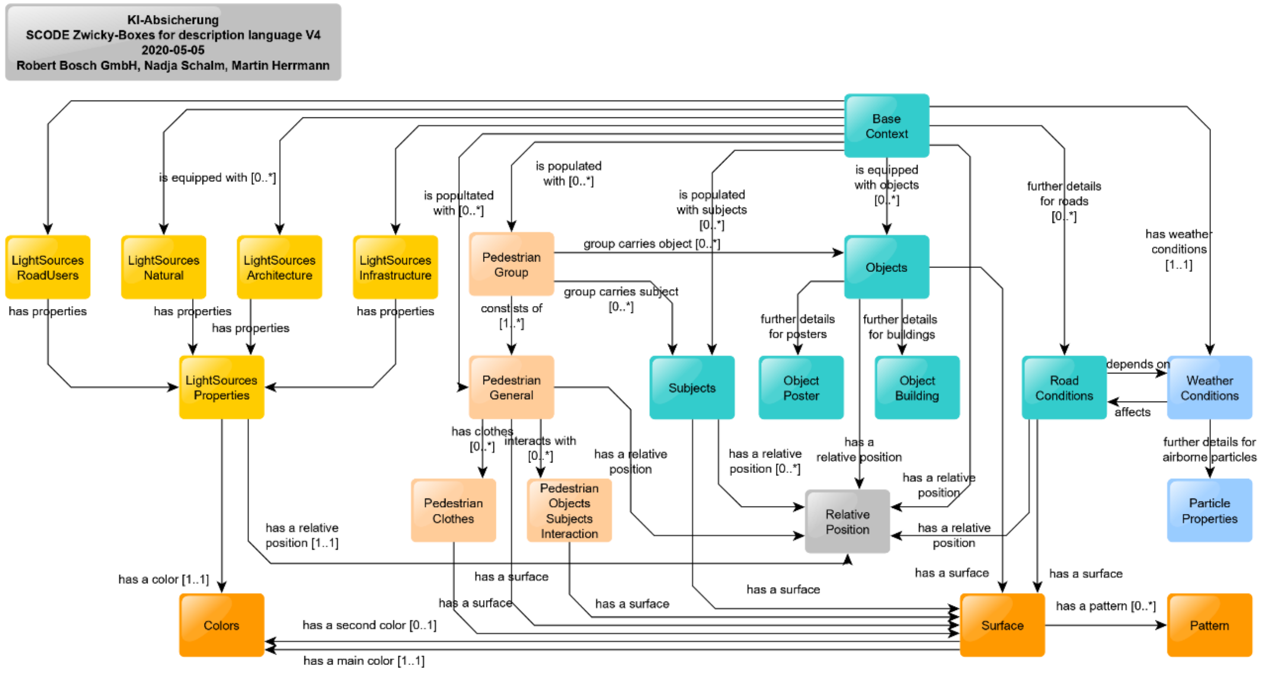
To specify light conditions the following approximate occurrences are possible: daylight sunny (20%), daylight cloudy (20%), daylight with backlight or cast shadows (20%), night (25%), twilight (10%), etc. Note that during the project, no night or twilight situations were modelled since no secondary light sources have been considered.

2.4.4 Result

2.4.4.1 Scenario description and scenario design

Zwicky boxes

Within TP4, an ontology for the ODD description has been developed using morphological analysis and Zwicky boxes. An overview of the developed exhaustive Zwicky box model is shown in the figure below (downloaded from https://confluence.vdali.de/x/6yV).



The following Zwicky boxes are part of this model (high prio dimension marked bold):

1. **Base context** (road map, junction type, pedestrian crossings, etc.)
2. **Objects** (vegetation, sky, building density, advertising pillar, etc.)
3. **Road conditions** (road surface, road quality, wetness, etc.)
4. **Object Building** (building material, height, shape, etc.)
5. Object Poster
6. Subjects

7. **Pedestrian Group** (number of persons per age, group density, etc.)
8. **Pedestrian General** (gender, body shape, height, pose, etc.)
9. **Pedestrian Clothes** (clothing type, upper part, lower part, shoes, etc.)
10. **Pedestrian Objects Subjects Interaction** (carried objects, person occluded, etc.)
11. **Relative Position Motion** (longitudinal/ lateral distance/ motion)
12. Weather Conditions
13. Particle Properties
14. Color
15. **Surface** (reflection, structure, wetness, etc.)
16. **Pattern** (no, stripes, patchy, etc.)
17. Light Sources Infrastructure
18. Light Sources Architecture
19. **Light Sources Natural** (sun elevation, sun sensor interaction, sky, etc.)
20. Light Sources Road Users
21. Light Source Properties

NCAP-like scenarios

We consider traffic situations with pedestrians, that are both difficult for perception and critical or near-critical for typical VRU warning and braking function. The situations are inspired by and similar to the Euro NCAP test protocol for AEB VRU systems [4]. To this end, we suggest to simulate sequences with MoCap pedestrian assets in the following scenarios:

1. Car-to-pedestrian nearside child (CPNC, see 7.2.6 in [4]) from obstruction: pedestrian crosses the street from right to left and behind obstructing parked car
2. same scenario as 1. but pedestrian crosses from left to right
3. Car-to-pedestrian longitudinal adult (CPLA, see 7.2.7 in [4]): pedestrian walks along the street looking away from the car
4. same scenario as 3. but pedestrian faces the car
5. Car-to-pedestrian turning adult (CPTA, see 7.2.8 in [4]): pedestrian crosses the street after car turns right at an intersection
6. same scenario as 5. but car turns left

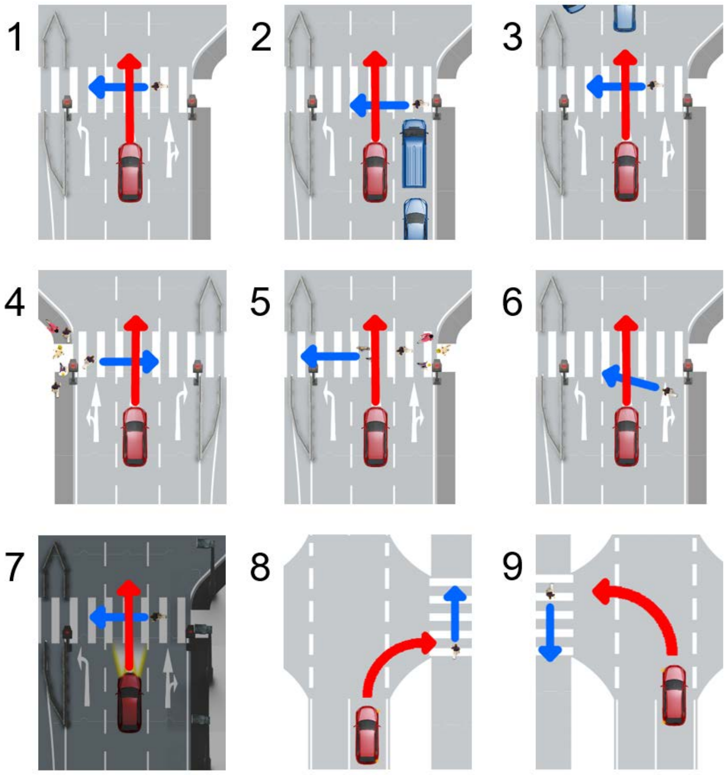
For test cases 1., 2., 5., and 6. at most three different timings between car and pedestrian trajectory should be simulated:

- Pedestrian starts to walk early and crosses the street just before the car passes by´.

- Pedestrian starts to walk such that there is a collision with the car (pedestrian asset should be removed as soon as it "cuts" a the virtual vehicle asset, i.e. distance between pedestrian and vehicle smaller than e.g. 2.5m).
- Pedestrian starts to walk late and the car passes by just before the pedestrian crosses the street.

The provided dataset already contains several similar exemplary situations, however, we believe an additional benefit can be generated by a systematic variation of the described test cases. Note that the occluding car scenario (1., 2.) especially enables the detection of a pedestrian in a safety-critical situation with different occlusion levels, whereas the turning car scenario (5., 6.) enables the detection of a pedestrian in a safety-critical situation in front of different background and in different positions in the image. The different timings between the car and pedestrian generally enables the detection of a pedestrian in a safety-critical situation at different distances and therefore different object sizes in the image.

A corresponding derivative result from P1 is given below.



No.	Base scenario	Light condition	Traffic lights	Vehicle velocity [km/h]	Pedestrian velocity [km/h]	NCAP scenario reference
1	Ego vehicle drives straight, pedestrian crosses intersection, no other vehicles	daylight	optional	20 -- 60	3 -- 5	CPFA / CPNA

No.	Base scenario	Light condition	Traffic lights	Vehicle velocity [km/h]	Pedestrian velocity [km/h]	NCAP scenario reference
2	Ego vehicle drives straight, pedestrian crosses intersection from right, other parked vehicles partially occlude the pedestrian	daylight	optional	20 -- 60	3 -- 5	CPNC
3	Ego vehicle drives straight, pedestrian crosses intersection from right, other vehicles in crossroads area	daylight	optional	20 -- 60	3 -- 5	
4	Ego vehicle drives straight, pedestrian group crosses intersection, no other vehicles	daylight	optional	20 -- 60	3 -- 5	
5	Ego vehicle drives straight, pedestrian steps out of pedestrian group and crosses intersection, no other vehicles	daylight	optional	20 -- 60	3 -- 5	
6	Ego vehicle drives straight, pedestrian crosses intersection outside crossing area, no other vehicles	daylight	optional	20 -- 60	3 -- 5	
7	Ego vehicle drives straight, pedestrian crosses intersection, no other vehicles	night with streetlights	optional	20 -- 60	3 -- 5	CPNA
8	Ego vehicle turns right, pedestrian crosses intersection after vehicle turns	daylight		10 -- 20	3 -- 5	CPTA
9	Ego vehicle turns left, pedestrian crosses	daylight		10 -- 20	3 -- 5	CPTA

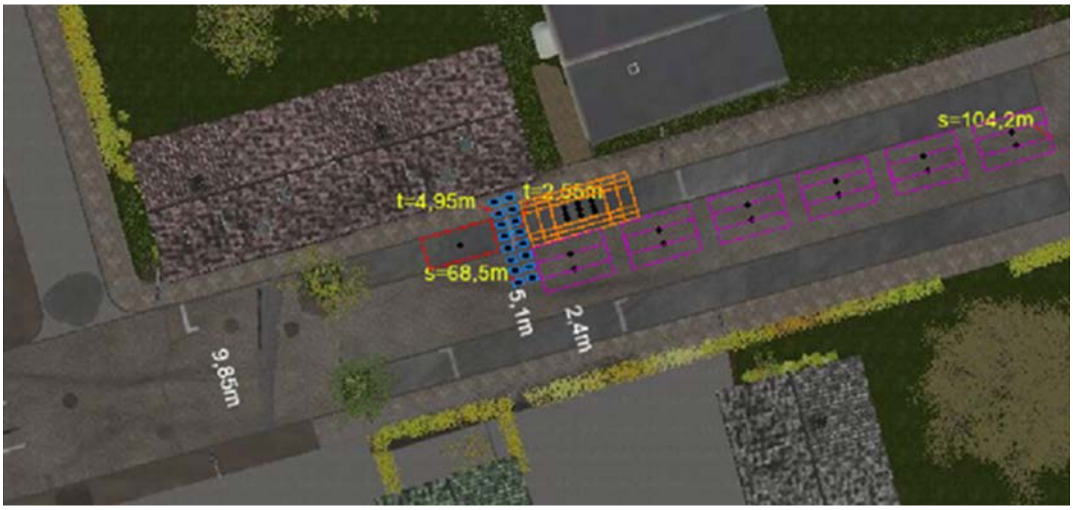
No.	Base scenario	Light condition	Traffic lights	Vehicle velocity [km/h]	Pedestrian velocity [km/h]	NCAP scenario reference
	intersection after vehicle turns					

Design of experiment

A specification of single frames for NCAP-like scenarios can be obtained using combinatorial testing and updated Zwicky boxes:

- Ego-vehicle position (18 states, marked magenta in image below)
- Pedestrian position (16 states, marked blue in image below)
- Pedestrian pose (walk stand, run, jump)
- Pedestrian asset (20 states)
- Pedestrian hip direction (8 states)
- Parked vehicle 1 type, color, position, color (2*5*9, marked orange in image below)
- Parked vehicle 2 type, color (2*5, marked red in image below)
- Illumination, sun direction and elevation (80 states)

The image below shows an example scenario that has been produced with the prototypical tool “ncap_to_json” that has been developed in AP 4.1. The approach has been aligned within P1 and has been provided by Mackevision in Tranche 6.



2.4.5 References

1. H. Caesar etal., "nuScenes: A multimodal dataset for autonomous driving," arXiv preprint arXiv:1903.11027, 2019.
2. A. Geiger, P. Lenz and R. Urtasun, "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite," CVPR, 2012.

3. J. Geyer etal., "A2D2: Audi Autonomous Driving Dataset," arXiv preprint arXiv:2004.06320, 2020.

4. Euro NCAP, "Test Protocol AEB VRU Systems," 2020. [Online]. Available: https://www.euroncap.com/de/fuer-ingenieure/protocols/vulnerable-road-user-vru-protection/

2.5 E1.2.5 Final: Definition von extremen Basisszenarien (zur Veröffentlichung)

2.5.1 Formal Classification

Criteria	Classification according to VHB
Type of result	*Document*
Group/Cluster	
Type of content	*Definition*
Classification level	*PU*

Authors: Simon Heming, Frederik Blank, Philipp Heidenreich, Michael Schuldes, Christopher Hauck

2.5.2 Description of the result

The overall report of E1.2.4 and E1.2.5 has been merged. Hence, the result here describes only one possible analysis of an extreme base scenario regarding the pose of the pedestrian. This scenario was chosen as one of the performance limiting factors identified in E1.2.6. Similar to the other scenarios this is a collaborative work between AP1.2, AP2.5, AP4.1 and AP4.4 coordinated within P1. The development and description of the experiment was mainly developed within AP1.2.

2.5.3 Approach

Poses are one of the identified performance limiting factors (PLFs), for which an exemplary dataset has been created, as well as planning for experiments, to validate the PLF and improve model performance on it.

Due to the data driven methods, with which DNNs are trained, the model needs to observe data during training, which are similar to those on which evaluation is performed later. For the use case of object detection, typical annotations of humans in form of 2D bounding boxes, are much higher than wide. Additionally, certain optical characteristics of humans, which are visible in walking pose, might not be visible for different poses. Hence, the detection of humans in untypical poses might be problematic for the network.

Studying the influence of poses on a normal dataset comes with certain limitations and difficulties, since other effects such as the contrast of the human, the relative size within the image or the degree of occlusion also have large effects on the detection performance e.g. what makes it difficult to correlate misdetections with pose-related influences.

The pose scenario was defined on the one hand to show how a PLF can be measured and on the other hand, how experiments can be designed to measure and reduce the limitation of the network. This pose scenario includes the definition of a dataset, which tries to enable an isolated evaluation of the effect of the pose. Additionally, the tooling is described, which was used to generate the dataset. Finally, experiment plans are described, which measure the effect of pose on the performance, and try to increase the performance for this particular problem.

This experiment extends the already defined experiments safety-oriented NCAP-like scenarios described in detail in AP4.1 and their evaluations in AP4.4.

2.5.4 Result

2.5.4.1 Dataset

Three different datasets are part of the pose scenario

1. A training dataset (4320 images)
2. A small test dataset, which can be used for a quick sanity check (180 images)
3. A (large) test dataset, on which the DNN performance can be evaluated (1728 + 3456 images)

The training dataset contains a diverse set of poses and scenes, to train a model capable of generalizing. The performance of a model can be initially evaluated on a small test dataset, which is similar in many aspects to the training dataset. The large test dataset then tests the performance and generalization capability of the model to a much greater extend.

To isolate the influence of the pose, different parameters are defined, which are either fix within a dataset, or define different variations. To create the dataset all combinations between the variable parameters are rendered. The parameters, which are varied either between or within a dataset are the following:

- The pedestrian asset
- The rotation of the asset to the ego camera
- The position of the asset in the base context
- The pose of the asset
- The sun position with regards to the relative azimuth and elevation
- The position of the ego camera within the base context

2.5.4.2 Training Dataset

For the trainings dataset the following parameters are used. The combinations of these possibilities creates a training dataset with a total of 4320 images.

- A single fixed pedestrian asset is used (male, with blue jeans and shirt)
- The asset takes three different rotations relative to the ego camera

- The asset is placed at 5 different positions at the same location in the base context
- The pedestrian takes 8 different poses (running1, laying1, kneeing, hugging, carrying, falling1, falling2, arms backward)
- 4 different sun positions are simulated
- Finally 9 different locations in the base context are selected, reflecting different scenarios during driving

2.5.4.3 Test 1 Dataset

This dataset does serve as a sanity check whether the model has any generalization capabilities and thus is similar to the training dataset in many aspects. The test set comes with a total of 180 total images.

- It contains a single asset, which is the same as the training asset, but has different clothing
- 1 rotation of the asset
- 5 positions
- 3 poses (running1, laying, falling1)
- 2 sun positions
- 6 locations

2.5.4.4 Test 2 Dataset

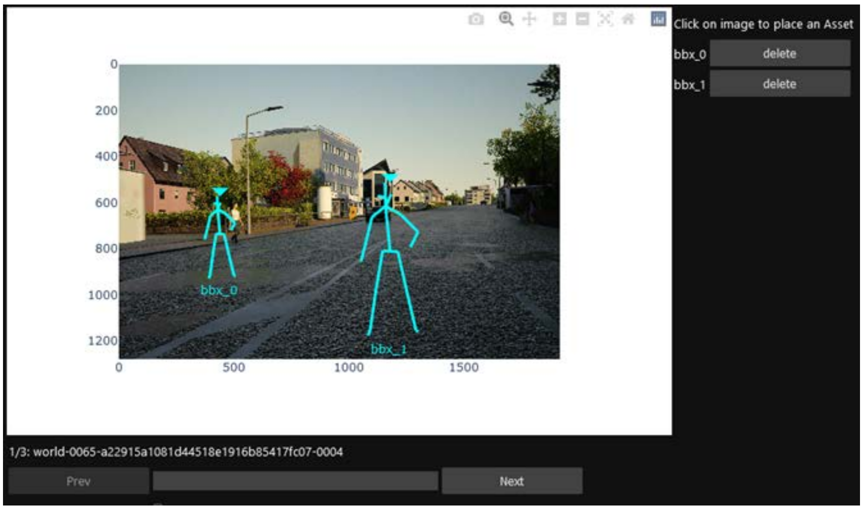
The second test dataset does not only change the pedestrian assets, but also uses different locations, with similar scenes in the base context. This can be used for a much more comprehensive analysis of the generalization capabilities of the model. The (large) test 2 dataset 1728 images for a male asset and 3456 images for female asset

- 2 different pedestrian assets (one male, same as test 1 dataset, one female, similar clothing to train dataset)
- 3 different rotations
- the male asset has 2 different rotations, the female asset has 4 different rotations
- 8 poses per pedestrian
- 4 sun positions
- 9 locations

2.5.4.5 Example gallery

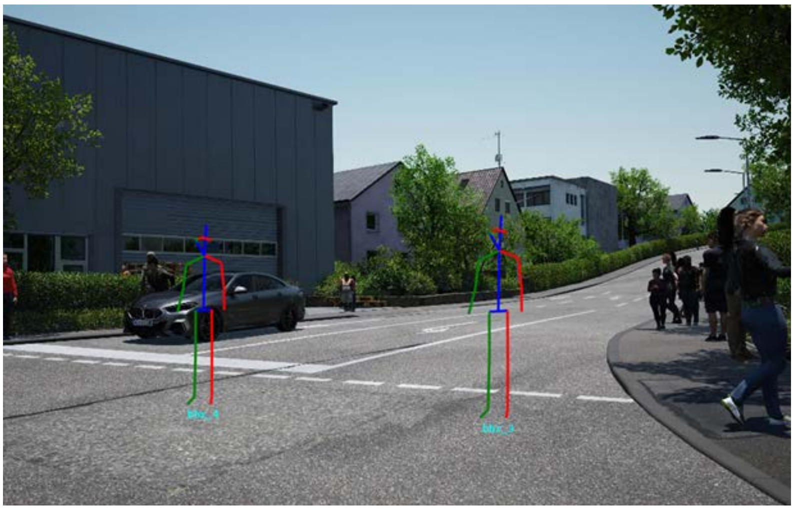
2.5.5 Tooling (by Michael Schuldes from E4.1.5 Final: Frontend für eine toolgestützte Datenanforderung und Eingabe/ Selektion von möglichen Parametervariationen)

The tool for creating a pedestrian pose dataset was integrated into the (NCAP-like) scenario request and parameter variation tool. It has a graphical user interface, that enables placement of pedestrians on images, definition of poses and creates simulation instructions from the set data. As an input, the images from simulation together with the respective meta annotations files are needed. The meta-annotation file is required to build the reference to the simluation world coordinate system. When the tool is started, the user is able to define a set of desired hip rotations of the desired pedestrian assets relative to the perspective of the camera. In addition the sun elevation and azimuth can be set. Trough a skeleton preview the asset MoCaP animation, animationtime and position are superimposed on the image from simulation. When exporting, for each combination of asset, position and rotation one simulation instruction file is created. The position and information on vehicles present in the simulated image are copied from the meta annotation file to the simulation instruction file.



Click on image to place an Asset

1/3: world-0065-a22915a1081d44518e1916b85417fc07-0004



2.5.5.1 Experiment planning

To evaluate the performance the following experiments have been planned.

The overall performance of the DNN is evaluated, by not only testing on the pose test sets, but also the test dataset of the original task (Tr#6 test). This is planned to ensure that the model does not only increase performance on the pose task, but the performance on the original task does not decrease. The following three experiments are designed and the rationale behind them is explained.

	Pre training data	Finetuning data	Tr #6 Testdata performance	Pose Test 1 dataset	Pose Test 2 dataset
Experiment 1	Tr#3+#4+#5+#6 Training data	None	Expect good performance	Expect bad performance	Expect bad performance
Experiment 2	Tr#3+#4+#5+#6 Training data	Pose train dataset	Expect worse performance than experiment 1	Expect better performance than experiment 1	Expect better performance than experiment 1

	Pre training data	Finetuning data	Tr #6 Testdata performance	Pose Test 1 dataset	Pose Test 2 dataset
Experiment 3	Tr#3+#4+#5+#6 Training data	Mix of pre training data and Pose train data	Comparable performance to experiment 1	Comparable performance to experiment 2	Comparable performance to experiment 2

2.5.5.2 General Expectation of the Experiment

Since the images often have the same background and not much asset variation a high amount of data augmentation required, to make sure that the model does not learn the pedestrian model.

The performance one some poses will be much better (e.g. on running pose likely very good), since it is similar to the walking pedestrians used mostly throughout Tr#3-6 training data. On the other hand the performance on poses with unusual bounding boxes (e.g. laying on the ground) should be evaluated carefully, since the asset will look much different, and have a wide but not very high bounding box. The effect of rotation of pedestrians laying on the ground might also have a huge influence, since the shape of the bounding box can vary drastically.

2.5.5.3 Experiment 1

To gain an initial baseline the performance is evaluated on the Tr#6 test data, as well as pose test 1 and 2 dataset. This model is the result of training on Tr#3-Tr#6 train data (train/val split). The expectation here is that the model would perform well on the Tr#6 test set, since it is similar to the training. However, since uncommon poses are only rarely part of the training data, the performance for the test sets is expected to be not very high.

2.5.5.4 Experiment 2

The model is now finetuned (the same model from experiment 1 is trained further) with the training dataset from the pose scenario. Here, the performance on pose test dataset should be much better. However, it can happen that performance decreases on the test data of Tr#6, since the fine-tuning only contains uncommon poses, which is not the only use-case of the model later on.

2.5.5.5 Experiment 3

The model is now finetuned (again, the same model from experiment 1) with the original training data as well as the pose training data. Each batch should contain some amount of data from each training set (e.g. 8 images from Tr#3-6 and 8 images from the pose train dataset). The evaluation should conclude that the performance does not decrease on the main task, while performance increases on the pose task.

2.6 E1.2.6 Final: Definition funktionaler Anforderungen und Einteilung in Fähigkeitsstufen (zur Veröfffentlichung)

2.6.1 Formal Classification

Criteria	Classification according to VHB
Type of result	*Document*
Group/Cluster	
Type of content	*Definition*
Classification level	*PU*

Authors: Simon Heming, Emil Schreiber , Frederik Blank , Andreas Albrecht

2.6.2 Description of the result

2.6.3 Approach

The aim of the E1.2.6 document is to formally define and specify the AI functions that will be developed in TP1 and that will be at the core of the investigations of this project.

This includes the following steps:

- outlining an abstract perception chain to facilitate a clear understanding of the terminology
- defining which part of the perception chain we will call the "AI function"
- defining the input and output interfaces of the AI function for our concrete applications
- setting basic requirements for the outputs that the AI function is supposed to deliver
- setting basic requirements for the inputs on which the AI function is supposed be able to generate resonable performance
- define attributes which influence the detection capabilites of the AI function

2.6.4 Result

2.6.4.1 The AI function and its location within the perception chain

2.6.4.1.1 Abstract perception chain

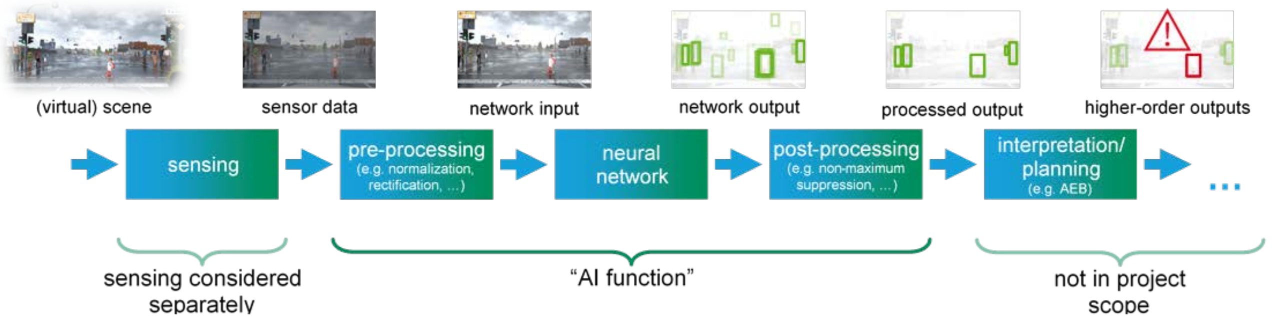

(virtual) scene
sensor data
network input
network output
processed output
higher-order outputs
sensing
pre-processing (e.g. normalization, rectification, ...)
neural network
post-processing (e.g. non-maximum suppression, ...)
interpretation/ planning (e.g. AEB)
...
sensing considered separately
"AI function"
not in project scope

***Figure 1:** Illustration of the considered perception chain with its main building blocks and intermediate interfaces. The images are shown as an example for a single-camera 2D-bounding-box detection, but any other application (e.g. pose estimation on combined camera and lidar data) would follow the same scheme.*

Here we want to define the perception chain in which the AI function is embedded. The perception chain processes information gathered from the (in our case virtual) world to produce some desired complex output.

The description here is intentionally abstract in order be applicable for a wide range of possible AI functions.

We split the perception chain into five main blocks: sensing, pre-processing, the neural network, post-processing, and interpretation or planning. In a real-world AI application the chain could continue further to produce further outputs of increasingly higher order, but this is not considered here.

2.6.4.1.1.1 Sensing

The sensing takes information from the (virtual) world and transforms it into sensor data. In our case the sensing will be modeled in the rendering pipeline with mathematical sensor models (for camera and lidar).

The output of the sensing block is the sensor data in the format specified (together with the annotation formats) in E1.2.3. Standardized processing that is already applied to the saved sensor data (e.g. conversion into a specific color space for image data) will be considered part of the sensor model.

2.6.4.1.1.2 Pre-processing

The optional pre-processing step includes any further processing of the sensor data that is done before feeding it into the neural network. This could include conversions of the data format, normalization of image data (in addition to a possible normalization that might already be included within a sensor model), or application of coordinate transformations like image rectification.

2.6.4.1.1.3 Neural network

This block encompasses the neural network itself that is at the core of our consideration.

2.6.4.1.1.4 Post-processing

The optional post-processing includes any essential processing that is applied to the raw network output in order to provide a more useful output for downstream applications.

A common example would be non-maximum suppression and thresholding for a bounding-box detection that serves to remove duplicate and low-relevance bounding boxes.

Post-processing would also include any data conversion that is used to bring the output into a common format for better comparability.

2.6.4.1.1.5 Interpretation/planning

All following higher-order processing that needs to happen to use the AI perception for fulfilling an intended high-level safety goal is considered collectively in this last block. This includes both interpretation (e.g. understanding which detected pedestrian is in danger of being hit) and planning (e.g. evaluating possible reactions) and could be further extended to following steps like acting.

These higher-order processing steps are not themselves part of the project scope, but they can be considered to provide context for the AI function.

2.6.4.1.1.6 Possible variations

Some neural-network applications investigated in this project might not directly fit a purely linear perception chain. In particular, sensor fusion models (e.g. those investigated in AP1.4) will include several parallel paths or multiple distinct neural networks. For these cases a clear separation into pre-processing, neural network and post-processing might no longer be possible, but the overall structure of the detection chain should remain valid.

2.6.4.1.2 Definition: “AI function”

Within this project we consider the “AI function” that will be the target of the assurance to be the combination of pre-processing, the actual neural network, and post-processing. The AI function takes the sensor data as its input and produces a low-level output.

Sensing is explicitly not included as part of the AI function. Instead, it is considered a part of the world. Variations of the sensors (e.g. sensing noise) will therefore be considered analogues to environmental influences (e.g. lighting conditions), rather than considering them on the same level as variations of the investigated neural network.

All higher-order processing steps beyond a simple post-processing, like interpretation and planning, are also not considered part of the AI function.

The AI function will be essentially defined by its input and output interfaces. Only AI functions that provide the same output can be directly compared. Therefore we want to define only a small number of possible output formats that all implemented AI functions should conform to.

AI functions operating on different inputs generally remain comparable: E.g. an AI function operating only on camera data can be compared with one operating only on lidar data by considering both as special cases of an AI function that operates on both camera and lidar data. Still, for the sake of simplicity we only want to consider a minimum amount of possible inputs.

2.6.4.2 Inputs

This section defines the possible input interfaces for the AI functions considered in the project. A specific AI function can use a single input or a combination of inputs.

For technical specifications we refer back to E1.2.1 (camera specifications), E1.2.3 (lidar specifications) and E1.2.3 (annotation formats) which also already includes suggestions for input data formats.

2.6.4.2.1 Mono camera

This sensor input provides RGB image frames from a single front-facing camera at the camera's full specified resolution and bit depth. The function can choose to consider only a reduced version of this data (e.g. half resolution, lower bit depth).

The parameters are taken from the camera sensor specification (E1.2.1). The data format for storage of the camera data is specified in the annotation format specifications (E1.2.3).

Table 1: Subset of camera senor data parameters (from E1.2.1).

parameter	value
resolution	1920 × 1280 (5.1 MP)
bit depth	32 bit (for EXR)
(all other camera parameters)	(also as specified in E1.2.1)

2.6.4.2.2 Lidar

Lidar sensor data input is provided as frames of point clouds. Following the specifications in annotation format document (E1.2.3), each point consists of its 3D coordinate (in the lidar coordinate system), a 3D velocity vector, the detection intensity, and polarization information in x- and y-direction.

According to the lidar sensor specification document (E1.2.2) it is foreseen that during the data generation process lidar data will be generated at a higher than realistic resolution and with a very large field of view. Realistic lidar data is then generated from this by downsampling and application of a lidar model. Only this final lidar sensor data is considered as the input of the AI function.

2.6.4.2.3 Meta information

The AI function can be provided with additional meta data as necessary.

Typically this will be static parameters that might influence the pre- or post-processing. An example would be camera parameters that can be used for applying a coordinate transformation in the pre-processing step.

It is not foreseen to provide dynamic meta information (e.g. odometry, weather information, time of day) as input for the AI function, but this can be included as a possible input if desirable. (Such information will still be available as ground-truth data and can potentially be used for KPIs and other evaluations.)

2.6.4.2.4 Single frame vs. sequence data input

Inputs, as well as outputs, of the AI function are considered to be a sequence of frames of data. I.e. the function is called with *N* data frames as input and will produce a sequence of output predictions.

While some detectors consider single-frame input, other applications might work on sequence data (in particular for AP1.4 and AP1.5). Single-frame functions can be trivially handled as multi-frame functions by applying repeatedly applying them to the individual frames of a longer sequence (e.g. to evaluate a tracking metric also for a single-frame detector).

If there are several input modalities (camera and lidar) the data frames should be synchronized.

2.6.4.3 Outputs and Performance Expectation

This section defines a small set of target outputs considered in this projects. For good comparability and to facilitate evaluation, testing and assurance, all AI functions should provide their output according these specification, which was consolidated in TP1 / TP3 Algorithm Output Formats.

The initial focus of assurance in TP4 will be on 2D bounding boxes (according to a poll conducted at the combined AP4.1/P1 and AP4.2/P3 workshop in Garching on 03.09.2019). AI functions that target other outputs (3D bounding boxes or pose) could optionally use post-processing to provide an additional (approximated) 2D bounding box output to be directly comparable with other 2D functions.

2.6.4.3.1 2D bounding boxes

The aim of an AI function targeting this output format should be to provide one 2D bounding box for each pedestrian that is at least partially visible in the camera image. Other object classes are not considered. It is not part of the AI function to consider the “importance” of a pedestrian (e.g. deciding whether the pedestrian is in the path of the ego vehicle).

2.6.4.3.1.1 Output details

- The bounding box should cover the full extent of the pedestrian, including parts that might be occluded or outside of the image frame. (E.g. if the legs of a pedestrian are covered by a car in the foreground, the output bounding box should estimate how far the legs extent behind the car.)

- Carried items (bags, rucksacks, etc.) are to be considered part of the pedestrian and included in the bounding box, whereas pushed or pulled objects (shopping carts, wheeled suitcases, etc.) are not (following the annotation specifications in E1.2.3 [1]).

- A 2D bounding box is given by its center coordinates and its width and height (in the image coordinate system).

- For each bounding box the output should include the network’s classification score (e.g. for generating ROC curves). This score does *not* need to be normalized to provide a reliable uncertainty estimate.

2.6.4.3.1.2 Basic requirements

The goal for the AI function will be to match the ground-truth bounding boxes "as well as possible", i.e. ideally a bounding box for each visible pedestrian, accurately estimated coordinates and dimensions, and no extra bounding boxes. The details of how these goals are quantified and weighted for optimizations with quantitative metrics will be specified in the KPI definitions of E1.2.7.

We set basic limitations for the bounding boxes that need to be detected (these are intentionally chosen ambitious in order to not exclude any potentially interesting corner cases, it is understood that the performance will be impacted before these limits are reached):

- Bounding boxes smaller than 33 px height can be ignored for training and evaluation.
- Bounding boxes with an occlusion of more than 80% can be ignored for training and evaluation.

2.6.4.3.2 2D semantic segmentation

The semantic segmentation predicts class labels for each pixel in the input image. In general a relatively large set of classes (e.g. 14: "person", "car", "road", "sky", etc.) will be used, so that pedestrian detection is only one subtask. If desired, the semantic segmentation output can however be evaluated as a pure pedestrian detection task by combining all non-pedestrian classes into a single background class.

The semantic segmentation does not aim to separate individual pedestrians (instance segmentation).

2.6.4.3.2.1 Output details

- The output of the semantic segmentation is a class map, mapping each pixel to the predicted class.
- Optionally confidence maps can be output that reflect the networks classification score for each pixel and each class. This score does *not* need to be normalized to provide a reliable uncertainty estimate.
- In case of reflections or transparencies the class of the primary object should be output (e.g. class "car" for a car window, even if you can see the sky through the window).

2.6.4.3.2.2 Basic requirements

For training and evaluation the semantic segmentation is generally evaluated on a per-pixel basis, based on counts of correctly and incorrectly labelled pixels. Metrics can take into account the relative frequency of classes.

Aspects like clustering of pixels of a certain class will generally not be considered for training (e.g. 10 wrongly labelled individual pixels spread over the image will be counted the same as 10 wrong connected pixels).

2.6.4.3.3 3D bounding boxes

This output format should provide a single 3D bounding box for each pedestrian that is at least partially visible for the camera and/or lidar. Other object classes are not considered.

2.6.4.3.3.1 Output details

(mostly the same as for 2D bounding boxes)

- The bounding box should cover the full extent of the pedestrian, including parts that might be occluded or outside of the image frame.
- Carried items (bags, rucksacks, etc.) are to be considered part of the pedestrian and included in the bounding box, whereas pushed or pulled objects (shopping carts, wheeled suitcases, etc.) are not (following the annotation specifications in E1.2.3 [1]).
- The 3D bounding box output is given by its center coordinates relative to the ego vehicle, its rotation (given as quaternion), and its size (height, width, depth).
- For each bounding box the output should include the network's classification score (e.g. for generating ROC curves). This score does *not* need to be normalized to provide a reliable uncertainty estimate.

2.6.4.3.3.2 Basic requirements

Like for 2D bounding boxes, the goal for the AI function will be to match the ground-truth bounding boxes "as well as possible", i.e. ideally a bounding box for each visible pedestrian, accurately estimated coordinates and dimensions, and no extra bounding boxes. The details of how these goals are quantified and weighted for optimizations with quantitative metrics will be specified in the KPI definitions of E1.2.7.

The orientation of bounding boxes can optionally be considered or ignored for training and evaluation.

We set basic limitations for the bounding boxes that need to be detected (these are intentionally chosen ambitious in order to not exclude any potentially interesting corner cases, it is understood that the performance will be impacted before these limits are reached):

- Bounding boxes beyond 100 m distance (outside of lidar range and small in camera view) can be ignored for training and evaluation.
- Bounding boxes with an occlusion of more than 90% (as seen from the camera or lidar) can be ignored for training and evaluation.

2.6.4.3.4 Other output modalities

The other output modalities considered for the project (e.g. 2D/3D pose estimation, intention recognition, etc.) are too varied to individually specify in detail here. Please refer to description of the respective results in AP1.3-5.

For comparative analysis and assurance most of these other targeted output modalities can additionally be converted (at least approximately) to pedestrian detections following the previously specified output formats (e.g. a 3D pose estimation can be converted to a 3D bounding box to be evaluated with the same metrics). This will, of course, only capture part of the capabilities of the respective AI function.

2.6.4.3.5 Performance expectations

Since the dataset is completely new and has a specific focus, it is difficult to work with values from other dataset to define an expected performance baseline.

The targeted metrics please refer to the metric collection: E1.2.7 Definition der Performanz- und Qualitäts-KPIs und ihrer Zielwerte, while baseline performance estimates and improvements should be taken from the results of AP1.3, AP1.4 and AP1.5.

Instead an operational design domain (ODD) is defined on which the AI should be able to operate.

2.6.4.4 Expectations for ODD coverage

The following is a draft for an operational design domain (ODD) in which the AI function(s) is expected to operate. It was developed in the P3 process (by Martin Schels , documented here) from a top-down perspective, defining desired customer-facing capabilities. Here we extend this by comments specifying in how far this is expected to be achievable from the perspective of the AI function developers.

2.6.4.4.1 Notes:

The label is meant to indicate ODD dimensions that are not expected to specifically cause any additional problems. There will still always be hard cases for the AI algorithms even within this "unproblematic" ODD.

The table was filled out trying to keep all planned AI algorithms in mind. However, there will be differences in how far the different algorithms are impacted by the individual factors.

(this table based on the ODD Definition: Function definition view)

Attribute	Sub-attribute	Sub-attribute	Desired capability	Expected achievability
Drivable area type	Autobahn		NO	
	Urban road		YES	NO PROBLEM
	Minor roads		NO	
Lane specification	number of lanes		YES minimum 2	NO PROBLEM
	lane dimensions		YES minimum 3.7m	NO PROBLEM
Drivable area geometry	Horizontal plane	Straight roads	YES	NO PROBLEM
		Curves	YES	NO PROBLEM

Attribute	Sub-attribute	Sub-attribute	Desired capability	Expected achievability
	Vertical plane	Level plane	YES	NO PROBLEM
		slope	YES	NO PROBLEM / CHALLANGING depending on slope
Drivable area surface type	Asphalt		YES	NO PROBLEM
	other		NO	
Vehicle velocity	<= 50km/h		YES	NO PROBLEM
	> 50km		NO	
Weather type	Sunny	Hot	YES	NO PROBLEM
		Cold	YES	NO PROBLEM
	Rainfall		NO	probably useful restrictions, making the problem significantly easier for the AI function
	Fog	depending on level of fog	YES / NO	NO PROBLEM / CHALLANGING / IMPOSSIBLE at some level of fog there will no detection possible
	Snow		NO	
Time of day	Daytime		YES	NO PROBLEM
	Night		NO	probably useful restriction, making the problem significantly easier for the AI function
VRU type	Pedestrian	Adult	YES	NO PROBLEM
		Child	YES	NO PROBLEM
		Elderly	YES	NO PROBLEM
	Cyclists		YES	CHALLANGING doable, but only with sufficient training data
	Skaters		NO	
VRU properties	Clothing		YES all kinds	NO PROBLEM / CHALLANGING (extremely unusual clothing could

Attribute	Sub-attribute	Sub-attribute	Desired capability	Expected achievability
				reduce performance, e.g. mascot costumes)
	Occlusion	minimum up to 80%	YES / NO	NO PROBLEM / CHALLANGING / IMPOSSIBLE at some level of occlusion there will be a significant performance impact, it might be useful to distinguish two or three levels of occlusion
	pose	walking (<=5km/h)	YES	NO PROBLEM
		running (>5km/h)	YES	NO PROBLEM
	Distance to ego vehicle	<= 20.6 m	YES	NO PROBLEM interaction with pedestrians this close can occur quickly and good detection performance is required
		> 20.6 m < 60m	YES	NO PROBLEM / CHALLANGING pedestrians that are further away become increasingly smaller and way harder to detect. also more chances for occlusion
		>= 60 m	NO	to be clear: we expect that there are still VRUs at >60m included in the data, but detection of these is not expected and evaluated
Surfaces	solid		YES	NO PROBLEM
	transparent		YES	NO PROBLEM / CHALLANGING transparent surfaces can potentially create challenging situations (e.g. should a person behind glass be detected or not?), but transparent surfaces should definitely be included in the ODD
	reflecting / wet road		YES	NO PROBLEM / CHALLANGING (same as for transparent surfaces)

Some of these attributes are considered challenging because they can take values that decrease the detection performance. To differentiate these attributes the performance limiting factors were defined, which are also investiaged throughout the project.

2.6.4.5 Performance Limiting factors

A performance limiting factor is a human comprehensible, physically or electronically measurable value, which directly or indirectly influences the performance of a DNN. The performance is expected to be significantly affected by the following defined factors. Some of them might be "tolerable" (e.g. by exclusion from ODD), others might be mitigated with sufficient specific training data (Andreas Albrecht , Frederik Blank). The focus of investigations within the project are the data specific PLFs, which can be mitigated by adjusting and optimizing the dataset itself, or the methods used during training.

2.6.4.5.1 Notes:

The table was filled out trying to keep all planned AI algorithms in mind, but with a slight focus on camera-based detections. We expect most listed factors to play a role for all algorithms. However, there will be differences in how far the different algorithms are impacted by the individual factors (e.g. a LIDAR-only 3D detector might not be significantly impacted by low color contrast between foreground and background objects). Some specific aspects for LIDAR-based detections are mentioned explicitly in the comments column.

Combinations of limiting factors (e.g. far way + strongly occluded) will compound the difficulty for the AI functions, making false detections more likely.

2.6.4.5.2 Categorization:

(Note that the following categories are not clear-cut.)

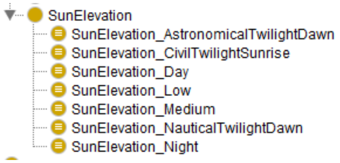
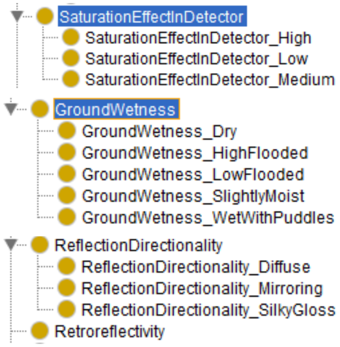
FUNDAMENTAL **Fundamental limitation:** A limitation that is caused by fundamental difficulties of the task set for the AI function. These limitation can either not be solved at all for the given sensor data (e.g. if strong fog completely obscures the recorded image, the limited field of view of the camera, too little resolution) or the AI function would have to develop very high-level reasoning skills (e.g. to reliably distinguish reflections of people from real people).

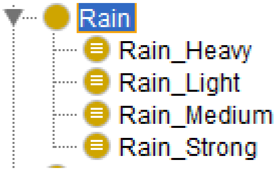
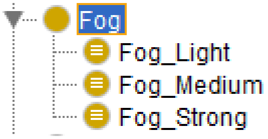
DATA-DRIVEN **Data-driven limitation:** A limitation that is mostly determined by the situation not being well represented (maybe also not easily representable) in the (expected) training data. Mitigating these limitations requires specific inclusion of sufficient relevant cases in the training data.

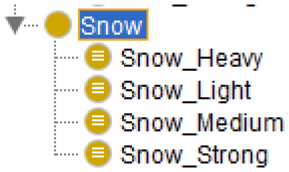
EXCLUDABLE **potentially addressable by excluding from ODD:** Limitations that could be mitigated by excluding them from the ODD. In the real world this could be done for situations that can be reliably detected up front and where it's acceptable that the system wont work (e.g. AI function not available during strong fog). In the project scope this can be further done by simple excluding such cases from the world that is considered.

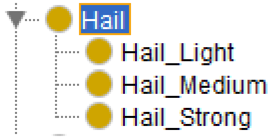
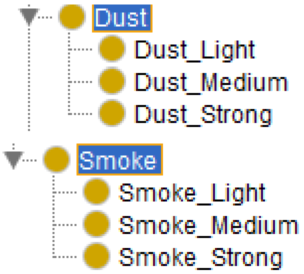
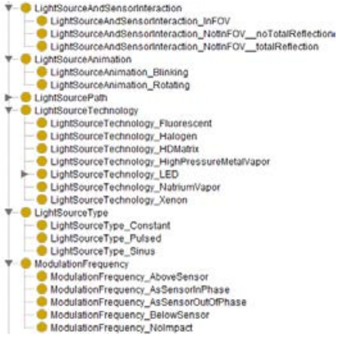
OUT-OF-DISTRIBUTION **potentially addressable by out-of-distribution detection:** Limitations that are easily accessible by methods like out-of-distribution detection.

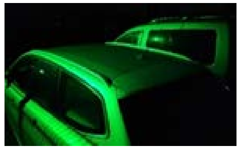
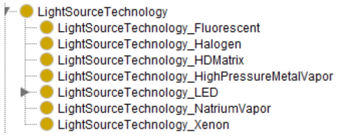
ACCEPTABLE **potentially acceptable:** Limitations that can be defined as acceptable because the will not have a negative impact on the derived driving decisions. Examples are detecting non human obstacles as human or not detecting pedestrians that are significantly outside the driving corridor.

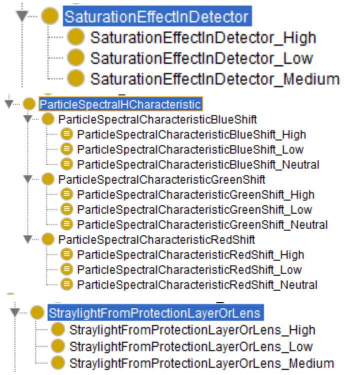
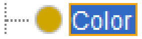
ID	limiting factor	Examples	Category	Problem for DNN	Problem of detection (data driven)	Problem for Camera	Problem for Lidar	Representation in ontology
1a	Low light intensity	Combination of dark background with low reflectance and a dark clothed pedestrian. Image source: [1]	FUNDAMENTAL (DATA-DRIVEN) (EXCLUDABLE) (OUT-OF-DISTRIBUTION)	No, the DNN has no problem by itself if the images from the sensor have no low light issues.	Difficulty of extracting features that are usually available in usual situations	Objects are not captured well by camera in lack of light intensity. Camera based detectors require a certain number of photons to leave the noise floor and reach a sufficient SNR (signal to noise ratio), otherwise DNN generates “weak features”		 SunElevation SunElevation_AstronomicalTwilightDawn SunElevation_CivilTwilightSunrise SunElevation_Day SunElevation_Low SunElevation_Medium SunElevation_NauticalTwilightDawn SunElevation_Night Color (all clothing parts have Individual color)
1b	Luminance high dynamic range	a combination of bright light sources (sun, flashlights headlamps..) in direct view or reflected from glossy surfaces (glass facades, wet road, metallic mirrors..) into the field of view and at the same time shadowed regions (by buildings, large trucks or vegetation)	FUNDAMENTAL (DATA-DRIVEN) (EXCLUDABLE) (OUT-OF-DISTRIBUTION)	No, the DNN has no problem by itself if the images from the sensor have no light issues.	Automotive systems are not often exposed to such situation, but as some of these situation appear systematically at night with wet roads or at sun rise and fall the relevance of such cases is not negligible and must be regarded in	At points with high dynamic luminance the objects such as road surface is not captured well by camera	There can be noise points caused by direct sunlight; noise points caused by some direct source of light (a wide wave spectrum, and that is why the sun shows this effect easily).	large variety of LightSources  SaturationEffectInDetector SaturationEffectInDetector_High SaturationEffectInDetector_Low SaturationEffectInDetector_Medium GroundWetness GroundWetness_Dry GroundWetness_HighFlooded GroundWetness_LowFlooded GroundWetness_SlightlyMoist GroundWetness_WetWithPuddles ReflectionDirectionality ReflectionDirectionality_Diffuse ReflectionDirectionality_Mirroring ReflectionDirectionality_SilkyGloss Retroreflectivity

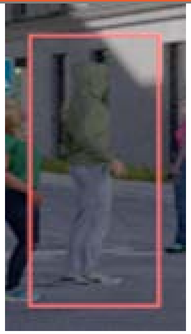
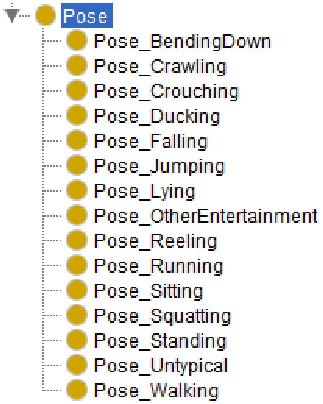
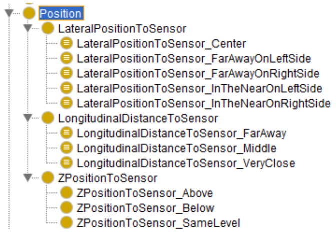
ID	limiting factor	Examples	Category	Problem for DNN	Problem of detection (data driven)	Problem for Camera	Problem for Lidar	Representation in ontology
					network training			
1c	Low contrast due to rain	Rainy weather conditions	FUNDAMENTAL (DATA-DRIVEN) (EXCLUDABLE) (OUT-OF-DISTRIBUTION)	DNN is unable to extract good features of objects with diluted contours if only camera sensor is used in rainy weather which leads to lower performance		a dilution of object contours and textures or change in color appearance in camera	If there is a blockage by rain distortion effects appear for Lidar, otherwise Lidar has a good performance in rain within 100 m and also for low hanging clouds	 Rain Rain_Heavy Rain_Light Rain_Medium Rain_Strong
1d	Low contrast due to fog	 an example of fog - image source: [1]	FUNDAMENTAL (DATA-DRIVEN) (EXCLUDABLE) (OUT-OF-DISTRIBUTION e.g. for blinding)	DNN is unable to extract good features of objects with diluted contours if only camera sensor is used in rainy weather which leads to lower performance	Lots of noise, which can change the method of detection. Change in the pattern or textures. could also lead to missing parts of the pedestrians, association error (leading to too small or also too large bounding boxes)	Dilution of textures or change in color appearance in camera	attenuation problem - Fog is a problem for Lidar - The lidar can detect surface with ice as a part of the road but snow and ice covered the host car -> big problem for the lidar	 Fog Fog_Light Fog_Medium Fog_Strong

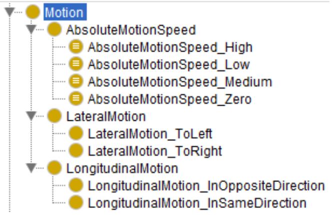
ID	limiting factor	Examples	Category	Problem for DNN	Problem of detection (data driven)	Problem for Camera	Problem for Lidar	Representation in ontology
1e	Low contrast due to snow			DNN is unable to extract good features of objects with diluted contours if only camera sensor is used in rainy weather which leads to lower performance	Lots of noise, which can change the method of detection. Change in the pattern or textures. could also lead to missing parts of the pedestrians, association error (leading to too small or also too large bounding boxes)	Dilution of textures or change in color appearance in camera	The lidar can detect surface with ice as a part of the road but snow and ice covered the host car -> big problem for the lidar Blockage A known case is when wet-snow adheres to the front cover and blocks the sensor. This happens very fast in some scenarios and the sensor is completely blind. Lower Temperatures - Lower temperatures (with rain, fog, snow or hail one usually has a lower temperature than in a summer sunny day) require a correct adjustment of the APD-	 Snow Snow_Heavy Snow_Light Snow_Medium Snow_Strong AdditionalRoadCoverType_Snow, RoadCoverMainType_Snow, GroundCoverMainType_Snow

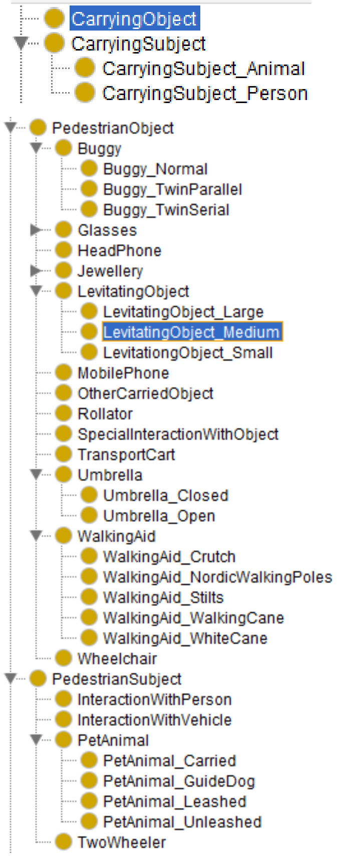
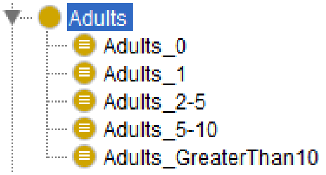
ID	limiting factor	Examples	Category	Problem for DNN	Problem of detection (data driven)	Problem for Camera	Problem for Lidar	Representation in ontology
							control, otherwise one may get more noise detection	
1f	Low contrast due to hail and falling hail	Bad weather condition, hail falling	FUNDAMENTAL (DATA-DRIVEN) (EXCLUDABLE) (OUT-OF-DISTRIBUTION)	No	lead to missing parts of the pedestrians, association error	a dilution of object contours and textures in camera	Lidar can detect the hail and falling hail however Behavior is not well known	 Hail Hail_Light Hail_Medium Hail_Strong AdditionalRoadCoverType_Hail, RoadCoverMainType_Hail, GroundCoverMainType_Hail
1g	Low contrast due to steam	Steam from manhole covers , Exhaust steam from other car	FUNDAMENTAL (DATA-DRIVEN) (EXCLUDABLE) (OUT-OF-DISTRIBUTION)	No	could lead to missing parts of the pedestrians, association error (leading to too small or also too large bounding boxes)	a dilution of object contours and textures in camera	The lidar can detect the steam as target but it makes a problem for the 3D cloud point areas of Lidar (noise distribution)	 Dust Dust_Light Dust_Medium Dust_Strong Smoke Smoke_Light Smoke_Medium Smoke_Strong
1h	Modulated light	saving infrastructure lights (street lamps, traffic lights), high efficient car lamps (LED head, tail and brake light), blue flashing light in conjunction with a signal tone e.g. ensures right of way for emergency or police cars	FUNDAMENTAL (DATA-DRIVEN) (OUT-OF-DISTRIBUTION e.g. for blinding)	No	As camera sensors are the only sensor technology being able to detect both signal light, color and situation context which is required to	modulation of the appearance of objects and even the generation of modulating light pattern on reflecting objects	No problem	 LightSourceAndSensorInteraction LightSourceAndSensorInteraction_InFOV LightSourceAndSensorInteraction_NotInFOV__noTotalReflection LightSourceAndSensorInteraction_NotInFOV__totalReflection LightSourceAnimation LightSourceAnimation_Blinking LightSourceAnimation_Rotating LightSourcePath LightSourceTechnology LightSourceTechnology_Fluorescent LightSourceTechnology_Halogen LightSourceTechnology_HDMatrix LightSourceTechnology_HighPressureMetalVapor LightSourceTechnology_LED LightSourceTechnology_NatriumVapor LightSourceTechnology_Xenon LightSourceType LightSourceType_Constant LightSourceType_Pulsed LightSourceType_Sinus ModulationFrequency ModulationFrequency_AboveSensor ModulationFrequency_AsSensorInPhase ModulationFrequency_AsSensorOutOfPhase ModulationFrequency_BelowSensor ModulationFrequency_NoImpact

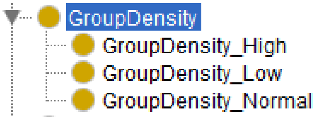
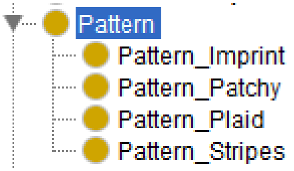
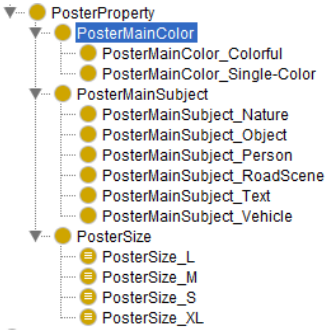
ID	limiting factor	Examples	Category	Problem for DNN	Problem of detection (data driven)	Problem for Camera	Problem for Lidar	Representation in ontology
					react in an appropriate way it is obvious that any interpretation error might lead to legal consequences or even risk of lives			BicycleLight, Brakelight, CameraFlashLight, CarriedLaser, CarriedPositionLights, EmergencyLight, HeadlightHighBeam, HeadlightLowBeam, SignalLight, TrafficEnforcementCameraFlashLight, TrafficLight, VariableTrafficSign
1i	Unbalanced spectrum	a narrow band light source (e.g. some local effective light sources as LED-flashlights of rescue vehicles, or “global scene effective” LED illuminants like “BauWatch green light” or high pressure sodium vapor light) is the dominant illuminant  effect of narrow band LED, image source [1]	(DATA-DRIVEN) (OUT-OF-DISTRIBUTION)	Can cause misdetection of objects due to change in appearance		changes the appearance of objects and the contrast against the background	No problem	 LightSourceTechnology LightSourceTechnology_Fluorescent LightSourceTechnology_Halogen LightSourceTechnology_HDMatrix LightSourceTechnology_HighPressureMetalVapor LightSourceTechnology_LED LightSourceTechnology_NatriumVapor LightSourceTechnology_Xenon

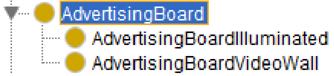
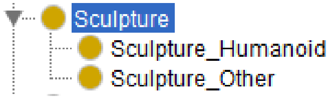
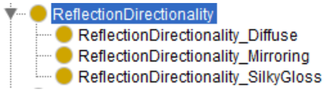
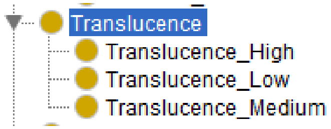
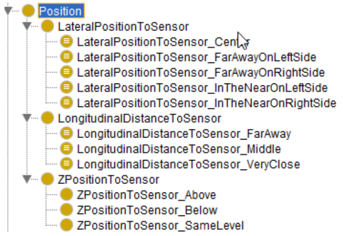
ID	limiting factor	Examples	Category	Problem for DNN	Problem of detection (data driven)	Problem for Camera	Problem for Lidar	Representation in ontology
1j	Light induced image artefacts	Example of light artefacts on the bus, image source [1]	(OUT-OF-DISTRIBUTION e.g. for blinding)	The part of objects with light reflections cannot be detected correctly		• changes the appearance of objects in the camera. • Depends where the object in the image + sun reflection.	• sun reflections only when the sun angels in crossing directly the lidar beam • Reflective markings make the detection difficult	 SaturationEffectInDetector SaturationEffectInDetector_High SaturationEffectInDetector_Low SaturationEffectInDetector_Medium ParticleSpectralHCharacteristic ParticleSpectralCharacteristicBlueShift ParticleSpectralCharacteristicBlueShift_High ParticleSpectralCharacteristicBlueShift_Low ParticleSpectralCharacteristicBlueShift_Neutral ParticleSpectralCharacteristicGreenShift ParticleSpectralCharacteristicGreenShift_High ParticleSpectralCharacteristicGreenShift_Low ParticleSpectralCharacteristicGreenShift_Neutral ParticleSpectralCharacteristicRedShift ParticleSpectralCharacteristicRedShift_High ParticleSpectralCharacteristicRedShift_Low ParticleSpectralCharacteristicRedShift_Neutral StraylightFromProtectionLayerOrLens StraylightFromProtectionLayerOrLens_High StraylightFromProtectionLayerOrLens_Low StraylightFromProtectionLayerOrLens_Medium
2	Low contrast, similar color to background	dark clothes on dark background or clothes of similar color to background		dark clothes on dark background will be hard for DNNs, but definitely need to be detected as well as possible (cannot be excluded from ODD)			No problem for lidar	 Color (all clothing parts have Individual color)

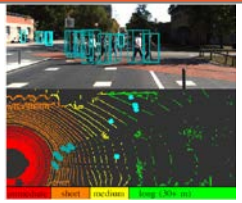
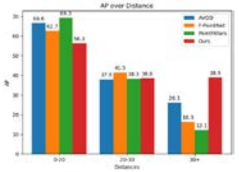
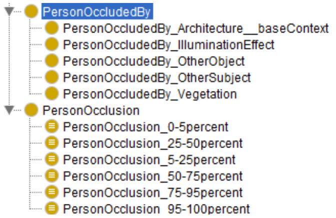
ID	limiting factor	Examples	Category	Problem for DNN	Problem of detection (data driven)	Problem for Camera	Problem for Lidar	Representation in ontology
								
3a	Uncommon person poses	• lying down (on the floor or on a bench) • handstand	DATA-DRIVEN	If there are not enough data available for training on uncommon poses, DNN will have difficulties in detection		Extreme body poses (e.g. upside-down persons) will most probably cause failure cases in pose estimation	Unusual scan shapes or less scan points might be a problem (ex: lying down)	 Pose Pose_BendingDown Pose_Crawling Pose_Crouching Pose_Ducking Pose_Falling Pose_Jumping Pose_Lying Pose_OtherEntertainment Pose_Reeling Pose_Running Pose_Sitting Pose_Squatting Pose_Standing Pose_Untypical Pose_Walking
3b	Uncommon person locations, above or below ground	• on a bridge or building (high above main ground plane) • in hole or ditch • on a strongly sloping road	DATA-DRIVEN ACCEPTABLE ?	• more likely to cause problems for 3D algorithms • needs training data • might not be very relevant as			No problem	 Position LateralPositionToSensor LateralPositionToSensor_Center LateralPositionToSensor_FarAwayOnLeftSide LateralPositionToSensor_FarAwayOnRightSide LateralPositionToSensor_InTheNearOnLeftSide LateralPositionToSensor_InTheNearOnRightSide LongitudinalDistanceToSensor LongitudinalDistanceToSensor_FarAway LongitudinalDistanceToSensor_Middle LongitudinalDistanceToSensor_VeryClose ZPositionToSensor ZPositionToSensor_Above ZPositionToSensor_Below ZPositionToSensor_SameLevel

ID	limiting factor	Examples	Category	Problem for DNN	Problem of detection (data driven)	Problem for Camera	Problem for Lidar	Representation in ontology
				long as such people are always outside of ego lane (but current metrics won't take this into account)				
3c	Uncommon person motion	- People on bikes - Wheelchairs - Scooters	DATA-DRIVEN EXCLUDABLE ?	- unlikely to be properly detected unless specifically included in training data - with sufficient training data these cases should be doable. - this depend on the image segmentation and ground truth of the training DNN				IsCarring_People, Wheelchair, TwoWheeler  Motion AbsoluteMotionSpeed AbsoluteMotionSpeed_High AbsoluteMotionSpeed_Low AbsoluteMotionSpeed_Medium AbsoluteMotionSpeed_Zero LateralMotion LateralMotion_ToLeft LateralMotion_ToRight LongitudinalMotion LongitudinalMotion_InOppositeDirection LongitudinalMotion_InSameDirection

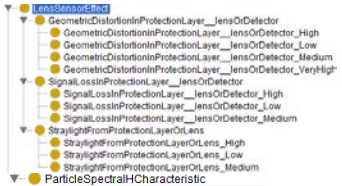
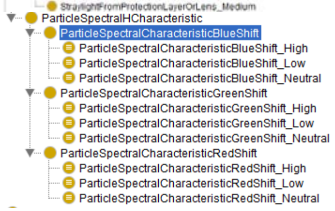
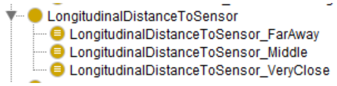
ID	limiting factor	Examples	Category	Problem for DNN	Problem of detection (data driven)	Problem for Camera	Problem for Lidar	Representation in ontology
3d	people in combination with large objects	• carrying large suitcase • carrying umbrella • pushing stroller/pram, shopping cart	DATA-DRIVEN	• distinction which objects to include or not to include within the bounding box/SemSeg class/etc. might be hard for network to learn • this depend on the image segmentation and ground truth of the training DNN				 CarryingObject CarryingSubject CarryingSubject_Animal CarryingSubject_Person PedestrianObject Buggy Buggy_Normal Buggy_TwinParallel Buggy_TwinSerial Glasses HeadPhone Jewellery LevitatingObject LevitatingObject_Large LevitatingObject_Medium LevitationgObject_Small MobilePhone OtherCarriedObject Rollator SpecialInteractionWithObject TransportCart Umbrella Umbrella_Closed Umbrella_Open WalkingAid WalkingAid_Crutch WalkingAid_NordicWalkingPoles WalkingAid_Stilts WalkingAid_WalkingCane WalkingAid_WhiteCane Wheelchair PedestrianSubject InteractionWithPerson InteractionWithVehicle PetAnimal PetAnimal_Carried PetAnimal_GuideDog PetAnimal_Leashed PetAnimal_Unleashed TwoWheeler
4	Groups of persons	• people walking together or hand in hand • people waiting at a bus stop • large crowd	FUNDAMENTAL (DATA-DRIVEN) ACCEPTABLE?	• this depend on the image segmentation and ground truth of the			less of a prbolem for LIDAR; depends on the occlusion level and scope of detection (individual	 Adults Adults_0 Adults_1 Adults_2-5 Adults_5-10 Adults_GreaterThan10

ID	limiting factor	Examples	Category	Problem for DNN	Problem of detection (data driven)	Problem for Camera	Problem for Lidar	Representation in ontology
				training DNN - close groups of people are likely to cause some false negatives - might be acceptable as long as group as a whole is still detected			instance or as a group)	same for Children, Teenagers, OldPersons InteractionWithPerson  GroupDensity GroupDensity_High GroupDensity_Low GroupDensity_Normal
5	Uncommon person clothing, strong patterns	- costumes - very strong patterns (dazzle camouflage or specifically designed to evade AI detection)	DATA-DRIVEN (FUNDAMENTAL) EXCLUDABLE	this depend on the image segmentation and ground truth of the training DNN		can we draw a line which kind of costumes cannot reasonably be expected to be detected as a person	Reflectivity of the material can affect the detection	 Pattern Pattern_Imprint Pattern_Patchy Pattern_Plaid Pattern_Stripes ClothingUpperPartType_Burka
6a	Persons depictions in on ads and posters	photos on advertisement posters	FUNDAMENTAL (DATA-DRIVEN) ACCEPTABLE ?	False positives when no lidar is used		causes false positives unless specifically trained for false detection might be acceptable here (we don't want to run over a billboard)	No problem for lidar. Lidar can detect 3D shapes of persons in advertisement	 PosterProperty PosterMainColor PosterMainColor_Colorful PosterMainColor_Single-Color PosterMainSubject PosterMainSubject_Nature PosterMainSubject_Object PosterMainSubject_Person PosterMainSubject_RoadScene PosterMainSubject_Text PosterMainSubject_Vehicle PosterSize PosterSize_L PosterSize_M PosterSize_S PosterSize_XL Pattern_Imprint, Poster,

ID	limiting factor	Examples	Category	Problem for DNN	Problem of detection (data driven)	Problem for Camera	Problem for Lidar	Representation in ontology
								 AdvertisingBoard AdvertisingBoardIlluminated AdvertisingBoardVideoWall
6b	Statues		FUNDAMENTAL (DATA-DRIVEN) ACCEPTABLE ?	False positives		will cause false positives unless specifically trained for false detection might be acceptable here (we don't want to run over a sculpture)	It will cause false positives. Lidar can detect 3D shapes of persons in advertisement	 Sculpture Sculpture_Humanoid Sculpture_Other
6c	Person reflections in specular surfaces	• reflection in glass façade • people inside building seen through window • people behind glass wall of bus stop • people in cars	FUNDAMENTAL (DATA-DRIVEN) ACCEPTABLE ?	consistent labeling essential for network (current data does not label people "occluded" by transparent objects)		reflections are likely to cause false positives unless specifically trained for (and even then there might be hard cases that in doubt should rather be detected as person)	Reflective surfaces can cause issues as points are introduced at wrong coordinates.	 ReflectionDirectionality ReflectionDirectionality_Diffuse ReflectionDirectionality_Mirroring ReflectionDirectionality_SilkyGloss  Translucence Translucence_High Translucence_Low Translucence_Medium PersonsVisible, IsCarring_People
7	Distant persons	People far away example results from DFKI:	FUNDAMENTAL ACCEPTABLE	performance will drop with distance of the detected pedestrians		Only few pixels in camera image. This is based on the range of the sensors (physical limitation rather than performance limitation)	Only a single scan line in the LIDAR data. Algorithms are affected depending on the distance, point density and height information of	 Position LateralPositionToSensor LateralPositionToSensor_Center LateralPositionToSensor_FarAwayOnLeftSide LateralPositionToSensor_FarAwayOnRightSide LateralPositionToSensor_InTheNearOnLeftSide LateralPositionToSensor_InTheNearOnRightSide LongitudinalDistanceToSensor LongitudinalDistanceToSensor_FarAway LongitudinalDistanceToSensor_Middle LongitudinalDistanceToSensor_VeryClose ZPositionToSensor ZPositionToSensor_Above ZPositionToSensor_Below ZPositionToSensor_SameLevel

ID	limiting factor	Examples	Category	Problem for DNN	Problem of detection (data driven)	Problem for Camera	Problem for Lidar	Representation in ontology
		 calculation by Philipp Heidenreich (see full calculations here: 1.2.1 Ergebnisbericht):  • 1.20m tall child at 60m distance → ~33px high bounding box at full camera resolution • most algorithms will use downsampled resolution (e.g. factor 4) making the above scenario already challenging					the distant objects.	
8	Occluded persons	• people seen through foreground vegetation, • only top of head visible above passing car	**FUNDAMENTAL**	• performance will drop with increasing occlusion • even small occlusions can make it hard to		few points available in camera	Lidar cannot see beyond solid objects. Hard to detect pedestrians if only few scan points available (ex. Body parts)	 PersonOccludedBy PersonOccludedBy_Architecture__baseContext PersonOccludedBy_IlluminationEffect PersonOccludedBy_OtherObject PersonOccludedBy_OtherSubject PersonOccludedBy_Vegetation PersonOcclusion PersonOcclusion_0-5percent PersonOcclusion_25-50percent PersonOcclusion_5-25percent PersonOcclusion_50-75percent PersonOcclusion_75-95percent PersonOcclusion_95-100percent

ID	limiting factor	Examples	Category	Problem for DNN	Problem of detection (data driven)	Problem for Camera	Problem for Lidar	Representation in ontology
				accurately estimate the full bounding box (which could lead to being counted as false detection in some metrics) • 80-90% occlusion might might already be close to impossible for some algorithms. • (also depending on which parts are occluded) • There could potentially be 99% occluded pedestrians that are still very relevant (e.g. about to enter the road				

ID	limiting factor	Examples	Category	Problem for DNN	Problem of detection (data driven)	Problem for Camera	Problem for Lidar	Representation in ontology
				behind a van) which the AI function cannot possibly detect. How to handle such cases in terms of performance metrics? • It is based on the training data and ground truth of the persons patterns				
9	Signal interferences						a problem only when the interferences crossing the beam directly above micro second --> rare situation till today	 LensSensorEffect GeometricDistortionInProtectionLayer__lensOrDetector GeometricDistortionInProtectionLayer__lensOrDetector_High GeometricDistortionInProtectionLayer__lensOrDetector_Low GeometricDistortionInProtectionLayer__lensOrDetector_Medium GeometricDistortionInProtectionLayer__lensOrDetector_VeryHigh SignalLossInProtectionLayer__lensOrDetector SignalLossInProtectionLayer__lensOrDetector_High SignalLossInProtectionLayer__lensOrDetector_Low SignalLossInProtectionLayer__lensOrDetector_Medium StraylightFromProtectionLayerOrLens StraylightFromProtectionLayerOrLens_High StraylightFromProtectionLayerOrLens_Low StraylightFromProtectionLayerOrLens_Medium  ParticleSpectralHCharacteristic ParticleSpectralCharacteristicBlueShift ParticleSpectralCharacteristicBlueShift_High ParticleSpectralCharacteristicBlueShift_Low ParticleSpectralCharacteristicBlueShift_Neutral ParticleSpectralCharacteristicGreenShift ParticleSpectralCharacteristicGreenShift_High ParticleSpectralCharacteristicGreenShift_Low ParticleSpectralCharacteristicGreenShift_Neutral ParticleSpectralCharacteristicRedShift ParticleSpectralCharacteristicRedShift_High ParticleSpectralCharacteristicRedShift_Low ParticleSpectralCharacteristicRedShift_Neutral
10	Person too close to	The person full body is not seen in the camera				The camera has a Field of View		 LongitudinalDistanceToSensor LongitudinalDistanceToSensor_FarAway LongitudinalDistanceToSensor_Middle LongitudinalDistanceToSensor_VeryClose

ID	limiting factor	Examples	Category	Problem for DNN	Problem of detection (data driven)	Problem for Camera	Problem for Lidar	Representation in ontology
	the camera					60° × 42.1° as described in E1.2.1. Considering that the pedestrian should be visible at least with height of H in camera, the minimum distance D of Pedestrian to the camera is calculate as D*tan(42.1/2)=H /2 = 1.3*H →this PLF is not relevant, because all pedestrians standing in front of the motor hood re fully seen		

[1]*Please refer to document provided for light Corner cases by Ulrich Seger (Robert Bosch GmbH) regarding the points in section 1a-j.

2.7 E1.2.7 Final: Definition der Performanz- und Qualitäts-KPIs und ihrer Zielwerte (zur Veröffentlichung)

2.7.1 Formal Classification

Criteria	Classification according to VHB
Type of result	*Document*
Group/Cluster	
Type of content	*Definition*
Classification level	*PU*

2.7.2 Description of the result

2.7.3 Approach

Based on literature research a number of metrics to characterize performance of deep neural network algorithms for pedestrian detection are collected, described and categorized in terms of their input modalities. The latter specifies applicability of a metric to a certain class of algorithm, e.g. availability of depth data, and the task this algorithm tries to solve.

2.7.4 Result

The partners provided this detailed metric information in a table (below) for various different tasks. Furthermore, for some metrics a reference implementation was developed and provided to the project.

Following table provides an overview of the metrics:

Identification					Classification				Details				
Metric ID	Metric	Category	Sub-category	Applicable task(s)	Time dependence	Description	Definition	Implementation	Benchmark	Remark	Contact	Reference Value	Importance regarding functional requirement
IDs ranging from MTRC-01000 to MTRC-01999	name of metric or collection of metrics	quality metric, resource metric, or something else?	further distinction	e.g. 2D bounding box detection	either calculated independently per frame, or depends on the history of frames	short description of what the metric measures	paper or website defining the metric	information about existing implementations (if available) or implementation in the project bitbucket	benchmark website/paper using this metric (if available)	e.g. "Why is this important?", "Will there be necessary adaptations?"	email of person who added the metric	Value or range of values with unit that a reference state of the art AI algorithm should reach with the given metric	A value that states the importance of this metric to measure the functional requirements specified in E1.2.6 Definition funktionaler Anforderungen und Einteilung in Fähigkeitsstufen 1 = low 2 = medium 3 = high
MTRC-01000	**Mean intersection over union (mIOU) or Jaccard Score**	quality metric	spatial localization quality	Semantic Segmentation	time-independent	The Jaccard Index or intersection over union is calculated as follows: true_positive / (true_positive + false_positive + false_negative). Usually used for semantic segmentation is the mean of the IoU over all	Shelhamer et. al.	https://luxproject.luxoft.com/stash/projects/KIA/repos/tp1_metric_script	http://www.cvlibs.net/datasets/kitti/eval_semseg.php?benchmark=semantics2015	Default performance measure for semantic-segmentation	korbinian.hagn@intel.com	>= 50% in a2d2	3

Identification				Classification					Details				
						classes leading to the mIoU							
MTRC-01001	**Sørensen-Dice score or F1-Score**	quality metric	spatial localization quality	Semantic Segmentation	time-independent	The Sørensen-Dice score (commonly called Dice-Score) or F1-Score: 2*true_positive / (2* true_positive + false_positive + false_negative). As with mIou usually the mean over all classes is used.	Bertels et. al.	https://luxproject.luxoft.com/stash/projects/KIA/repos/tp1_metric_script		IOU vs. Dice-Score	korbinian.hagn@intel.com	>= 0.5	3
MTRC-01002	**log-average miss rate (following Caltech PDET evaluation scheme)**	quality metric	detection/ classification	2D bounding box detection	time-independent	Miss rate (= FN/(TP+FN)) averaged over several operating points with different FPPI rate (false positives per image) equidistant on a log scale.	Dollár et al. 2012	reference implementation available in Matlab	Caltech Pedestrian Detection Benchmark	The evaluation protocol includes well-thought-out concepts of how exactly to count detections based on overlap, occlusion, multi-detections, etc.	emil.schreiber@de.bosch.com	good values are roughly: <0.30 (for peds. ≥100px) <0.55 (for peds. ≥50px)	3
MTRC-01003	**Average Precision (AP)**	quality metric	detection/ classification	2D bounding box detection	time-independent	The precision/recall curve is computed from a method's ranked output. AP is definded as the interpolated	Salton and McGill 1986, PASCAL VOC, COCO	https://github.com/cvgroup-njust/CityPersons/blob/master/evaluation/eval_script/coco.py	https://paperswithcode.com/sota/object-detection-on-coco	-	patrick.feifel@opel-vauxhall.com	-	3

Identification					Classification				Details				
						mean between both curves.							
MTRC-01004	**Average 2D localization precision**	quality metric	spatial localization quality	2D bounding box detection	time-independent	Measure the shift between ground truth and prediction bounding box	-	MTRC-01004 and MTRC-01026: 2D localization precision	-	Important to further calculate the error in world coordinate system. Please also see MTRC-01026.	christian.hellert@continental-corporation.com	-	2
MTRC-01005	**SRT metric (scaling, rotation, translation error)**	quality metric	spatial localization quality	3D bounding box detection	time-independent	-	Simon et al. 2019		-	-	timo.saemann@valeo.com	-	2
MTRC-01006	**nuScenes detection score (NDS)**	quality metric	spatial localization quality	3D bounding box detection	time-independent	A weighted sum of mAP, mean Average Translation Error, mean Average Scale Error, mean Average Orientation Error, mean Average Velocity Error and mean Average Attribute Error	Caesar et al. 2019	nuscenes-devkit	nuScenes Detection Task	-	david_michael.fuerst@dfki.de	-	2
MTRC-01007	**Birds-Eye-View mean Average Precision (BEV mAP)**	quality metric	detection/ classification	3D bounding box detection	time-independent	Works like normal Average Precision, only that the Positives are determined using Birds Eye View IoU and not image view IoU.	Simonelli et al. 2019	kitti devkit	KITTI BEV	Most popular / used for 3d Detection	david_michael.fuerst@dfki.de	-	2

Identification				Classification					Details				
MTRC-01008	**3D mean Average Precision (3D mAP)**	quality metric	detection/ classification	3D bounding box detection	time-independent	Works like normal Average Precision, only that the positives are determined with 3D IoU and not image view IoU.	Simonelli et al. 2019	kitti devkit	KITTI 3D	-	david_michael.fuerst@dfki.de	--	3
MTRC-01009	**Mean Per Joint Position Error (MPJPE)**	quality metric	spatial localization quality	3D pose estimation	time-independent	Computes the euclidian position error and averages it over all joints.	Ionescu et al. 2014		Human3.6M	Most popular for 3d pose estimation	nikolas.brasch@tum.de	-	3
MTRC-01010	**Mean Per Joint Angle Error (MPJAE)**	quality metric	spatial localization quality	3D pose estimation	time-independent		Ionescu et al. 2014		Human3.6M	-	nikolas.brasch@tum.de	-	3
MTRC-01011	**Mean Per Joint Localization Error (MPJLE)**	quality metric	spatial localization quality	3D pose estimation	time-independent		Ionescu et al. 2014		Human3.6M	-	nikolas.brasch@tum.de	-	3
MTRC-01012	**Object Keypoint Similarity (OKS)**	quality metric	spatial localization quality	3D pose estimation, 2D pose estimation	time-independent	“To compute OKS, we pass the di[stance] through an unnormalized Guassian with standard deviation [...], where s is the object scale and κi is a per-keypont constant that controls falloff.” Then these are averaged over all annotated keypoints.	COCO Keypoint Evaluation		COCO	-	david_michael.fuerst@dfki.de	-	2

Identification				Classification					Details				
MTRC-01013	**Percentage of Correct Keypoints (PCK)**	quality metric	detection/ classification	3D pose estimation, 2D pose estimation	time-independent	Measure the percentage of keypoints that have less than a threshold deviation from ground truth.	Yang et al. 2012		LSP & MPII HP	-	david_michael.fuerst@dfki.de	-	2
MTRC-01014	**PCK 50% head as threshold (PCKh)**	quality metric	detection/ classification	3D pose estimation, 2D pose estimation	time-independent	Measure the percentage of keypoints that have less than a 50 % head size deviation from ground truth.			LSP	-	david_michael.fuerst@dfki.de	-	2
MTRC-01015	**CLEARMOT (MOTA)**	quality metric	detection/ classification	Tracking	time-dependent	1-Error in missed sequence, false positives, missmatches	Bernardin et al. 2008	KITTI tracking development kit	KITTI Multi Target Tracking	-	david_michael.fuerst@dfki.de	-	2
MTRC-01016	**Mostly Tracked(MT) /Partially Tracked (PT)/Mostly Lost (ML)**	quality metric	detection/ classification	Tracking	time-dependent	Mostly Tracked = >80% of time tracked, Mostly Lost = <20% tracked, Partialy Tracked = else	Li et al. 2009	KITTI tracking development kit	KITTI Multi Target Tracking	-	david_michael.fuerst@dfki.de	-	2
MTRC-01017	**Inference time**	resource metric	inference	any		The duration of a forward-pass, i.e. the amount of time for a network to compute its output. Fast and guaranteed inference time is critical for detecting safety-relevant classes	-	-	-	-	jan.david.schneider@volkswagen.de	<= 0.3s for semantic segmentation	1

Identification				Classification					Details				
MTRC-01018	**Memory consumption**	resource metric	inference	any		Usage of RAM / GPU-RAM. Not safety relevant, but a technical requirement	-	-	-	-	jan.david.schneider@volkswagen.de	-	1
MTRC-01019	**Network complexity**	resource metric	inference	any		Complexity of the network in terms of computations that are needed. FLOPs, number of sequential nodes / layers during inference, etc. Not safety relevant, but a technical requirement.	-	-	-	-	jan.david.schneider@volkswagen.de	-	1
MTRC-01020	**Network size**	resource metric	training	any		Number of parameters. Not safety relevant, but a technical requirement.	-	-	-	-	jan.david.schneider@volkswagen.de	-	1
MTRC-01021	**Convergence**	resource metric	training	any		How many epochs / time takes the model to converge.	-	-	-	-	n/a	-	1
MTRC-01022	**Frequency Weighted Intersection over Union (FWIoU)**	quality metric	spatial localization quality	Semantic Segmentation	time-independent	Assings a weight [0,1] to the classes according to the area that a certain class covers in the ground truth, then calculates the IoU for each class and sums up the weighted	Hoffmann et al. 2017	Source of sem_seg_metric_calculation.py - tp1_metric_script - LUXproject Bitbucket (luxoft.com)	-	-	korbinian.hagn@intel.com	>= 90% in a2d2	3

Identification				Classification					Details				
						IoUs. A.K.A. frequency weighted intersection over union, fwIoU							
MTRC-01023	**Pixel Accuracy**	quality metric	spatial localization quality	Semantic Segmentation	time-independent	Calculates the accuracy of the neural network. I.e. the sum of true positive and true negative values over the sum of all pixels (true negative + true positive + false positive + false negative).	-	Source of sem_seg_metric_calculation.py - tp1_metric_script - LUXproject Bitbucket (luxoft.com)	-	-	korbinian.hagn@intel.com	>= 90% in a2d2	2
MTRC-01024	**Mean Pixel Accuracy**	quality metric	spatial localization quality	Semantic Segmentation	time-independent	Calculates the accuracy of the neural network for each class, then takes the mean over these pixel accuracys.	-	Source of sem_seg_metric_calculation.py - tp1_metric_script - LUXproject Bitbucket (luxoft.com)	-	-	korbinian.hagn@intel.com	>= 50% in a2d2	2
MTRC-01025	**FLOPS**	resource metric	inference	any	time-independent	Number of floating point operations needed for a forward pass of the model	-	-	-	-	adrian.loy@merantix.com	-	1
MTRC-01026	**2D localization precision distribution**	quality metric	spatial localization quality	2D bounding box detection	time-independent	Calculating the joint & marginal distribution of prediction bounding box localization error	-	MTRC-01004 and MTRC-01026: 2D localization precision	-	Please also see MTRC-01004.	iwo.kurzidem@iks.fraunhofer.de	-	3

Identification					**Classification**				**Details**				
MTRC-01027	**Precision**	quality metric	detection/ classification	2D bounding box detection	time-independent	This metric will be calculated on a per frame basis as well as an average over the entire evaluated dataset.	-	by Simon Heming for OpelSSD: https://luxproject.luxoft.com/stash/projects/KIA/repos/tp1_ap1.3_2d-bounding-box_ssd/browse/ssd/metrics/IOU_instance_evaluation/lib/iou_instance_evaluator.py	-	-	Simon.Heming@de.bosch.com	-	3
MTRC-01028	**Recall**	quality metric	detection/ classification	2D bounding box detection	time-independent	This metric will be calculated on a per frame basis as well as an average over the entire evaluated dataset.	-	by Simon Heming for OpelSSD: https://luxproject.luxoft.com/stash/projects/KIA/repos/tp1_ap1.3_2d-bounding-box_ssd/browse/ssd/metrics/IOU_instance_evaluation/lib/iou_instance_evaluator.py	-	-	Simon.Heming@de.bosch.com	-	3
MTRC-01029	**Number of "True Positives"**	quality metric	detection/ classification	2D bounding box detection	time-independent	Counting statistic. This metric will be calculated on a per frame basis as well as an average over the entire	-	by Simon Heming for OpelSSD: https://luxproject.luxoft.com/stash/projects/KIA/repos/tp1_ap1.3_2d-	-	-	Simon.Heming@de.bosch.com	-	3

Identification				Classification					Details				
						evaluated dataset.		bounding-box_ssd/browse/ssd/metrics/IOU_instance_evaluation/lib/iou_instance_evaluator.py					
MTRC-01030	**Number of "False Positives"**	quality metric	detection/ classification	2D bounding box detection	time-independent	Counting statistic. This metric will be calculated on a per frame basis as well as an average over the entire evaluated dataset.	-	by Simon Heming for OpelSSD: https://luxproject.luxoft.com/stash/projects/KIA/repos/tp1_ap1.3_2d-bounding-box_ssd/browse/ssd/metrics/IOU_instance_evaluation/lib/iou_instance_evaluator.py	-	-	Simon.Heming@de.bosch.com	-	3
MTRC-01031	**Number of "False Negatives"**	quality metric	detection/ classification	2D bounding box detection	time-independent	Counting statistic. This metric will be calculated on a per frame basis as well as an average over the entire evaluated dataset.	-	by Simon Heming for OpelSSD: https://luxproject.luxoft.com/stash/projects/KIA/repos/tp1_ap1.3_2d-bounding-box_ssd/browse/ssd/metrics/IOU_instance_evaluation/lib/iou_instance_evaluator.py	-	-	Simon.Heming@de.bosch.com	-	3

Identification				Classification					Details				
MTRC-01032	**Number of "Ignored Predictions"**	quality metric	detection/ classification	2D bounding box detection	time-independent	Counting statistic. This metric will be calculated on a per frame basis as well as an average over the entire evaluated dataset. The predestrian can be set to be ignored in the evaluation metrics, instead it will show up here.	-	by Simon Heming for OpelSSD: https://luxproject.luxoft.com/stash/projects/KIA/repos/tp1_ap1.3_2d-bounding-box_ssd/browse/ssd/metrics/IOU_instance_evaluation/lib/iou_instance_evaluator.py	-	-	Simon.Heming@de.bosch.com	-	3
MTRC-01033	**Number of "Exclude GT labels"**	quality metric	detection/ classification	2D bounding box detection	time-independent	Counting statistic. This metric will be calculated on a per frame basis as well as an average over the entire evaluated dataset. The number of pedestrians that should be ignored for this frame	-	by Simon Heming for OpelSSD: https://luxproject.luxoft.com/stash/projects/KIA/repos/tp1_ap1.3_2d-bounding-box_ssd/browse/ssd/metrics/IOU_instance_evaluation/lib/iou_instance_evaluator.py	-	-	Simon.Heming@de.bosch.com	-	3
MTRC-01035	**Multiply-accumulate operation**	resource metric	inference	any	time-independent	measures the number of operatation of one inference. MAC is round about coralating to the interence time and serves as an indicater	https://rdcu.be/b4NYT	-	-	-	lydia.gauerhof@de.bosch.com	-	2

Identification				Classification					Details				
						for a rough estimation of computation time demand.							
MTRC-01036	**Minimum Time between False Positives**	quality metric	detection/ classification	any	time-dependent	Measure the time (in milliseconds) between false positives in consecutive frames.	-	-	-	-	christian.hellert@continental-corporation.com	-	2
MTRC-01037	**Distribution of Time between False Positives**	quality metric	detection/ classification	any	time-dependent	Calculate the distribution of times between false positives	-	-	-	-	christian.hellert@continental-corporation.com	-	2
MTRC-01038	**Minimum Time between False Negatives**	quality metric	detection/ classification	any	time-dependent	Measure the time (in milliseconds) between false negatives in consecutive frames.	-	-	-	-	christian.hellert@continental-corporation.com	-	2
MTRC-01039	**Distribution of Time between False Negatives**	quality metric	detection/ classification	any	time-dependent	Calculate the distribution of times between false negatives	-	-	-	-	christian.hellert@continental-corporation.com	-	2
MTRC-01040	**Precision-Recall curve**	quality metric	detection/ classification	2D bounding box detection	time-independent	Precision and recall evaluated at different classification thresholds	-	-	-	-	christian.hellert@continental-corporation.com	-	3

2.8 E1.2.8 Final: Erweiterung des Spezifikationsdokumentes zu Szenarien-Beschreibungssprache (zur Veröffentlichung)

2.8.1 Formal Classification

Criteria	Classification according to VHB
Type of result	*Document*
Group/Cluster	
Type of content	*Specification*
Classification level	*PU*

2.8.2 Description of the result

The aim of E1.2.8 was to extend the scenario description language (also called ontology) that has been developed in AP4.1 (especially E4.1.4a and E4.1.4b) in order to accommodate specific requirements in AP1.2, in particular derived from E1.2.4 and E1.2.5. These regularly checked extensions should ensure that the scenario description language is always capable of capturing all aspects of the scenarios that are important for training data.

2.8.3 Approach

The definition and specification of these former mentioned "important scenarios" was done in E1.2.6. In addition to the definition of requirements for input and output for the AI function, attributes which influence the detection capabilites of the AI function were defined. These attributes are the performance limiting factors (PLFs).

Considering these performance limiting factors E1.2.4 defined base scenarios for synthetic datatsets. The final report of E1.2.5 describes one example scenario regarding the pose of the pedestrian. For this pose scenario an exemplary dataset has been created, as well as planning for experiments, to validate the PLF and improve model performance on it. The pose scenario was defined on the one hand to show how a PLF can be measured and on the other hand, how experiments can be designed to measure and reduce the limitation of the network.

Describing these base scenarios can be done with the scenario description language or ontology developed in E4.1.4a and E4.1.4b. Based on the rules of the ontology you can consistently define scenarios and specify and request images for training data. In E1.2.8 we checked if the performance limiting factors elaborated within E1.2.6 (and basic modules of the base scenarios) can be described with this ontology. We tried to describe all performance limiting factors by the available ontology and extended the given table from E1.2.6 by a column "representation in ontology": PLFs table. Feedback was given to E4.1.4 and necessary changes were discussed, so that the most of the PLFs could be captured by the ontology.

The last step to an applicable result was to develop a tool to create simulation requests in the format of the scenario description language. The front-end developed by E4.1.5 enables selection and combination of PLFs and supports the creation of scenes defined through the machine-readable format. With this tool datasets for the base scenarios defined in E1.2.4 can easily be required.

2.8.4 Result

Being able to capture the performance limiting factors is an important goal for the final scenario description language/ontology to describe the developed base scenarios. In this report the close transition between collaborating sub-workpackages is highlighted. The way from defining performance limiting factors to the representation of these in a synthetic dataset.

3 AP1.3 Kamera-basierte Algorithmen

E1.3.1 Final: nur projektintern für KI Absicherung verfügbar

E1.3.2 Final: nur projektintern für KI Absicherung verfügbar

E1.3.3a Final: nur projektintern für KI Absicherung verfügbar

E1.3.3b Final: Implementierung der funktionalen Algorithmen: Pixel-akurate semantische Segmentierung (zur Verbreitung innerhalb der KI-Familie, keine Veröffentlichung)

3.1.1 Formal Classification

Criteria	Classification according to VHB
Type of result	*Code*
Group/Cluster	
Type of content	*DL Model*
Classification level	*INT, LI*

3.1.2 Approach

Based on literature research a number of state-of-the-art deep neural network algorithms for pedestrian detection are selected. This result implements a pixel-accurate semantic Segmentation. Semantic segmentation performs a pixelwise classification of images. This way, a precise classification of each pixel or, in other words, segmentation of the image into classes can be performed. In the KIA Dataset, a special weight is placed on the segmentation of pedestrians.

The chosen algorithms for semantic-segmentation are DeeplabV3+ (Intel) and DeeplabV3 (ZF). The algorithms were trained on real-world and our synthetic data, evaluated and subsequently provided to the project in terms of code and trained weights.

3.1.2.1 DeeplabV3+

DeeplabV3+ by Google is using atruous convolution and an encoder decoder network and Feature Pyramid Pooling (FPP) for feature map creation:

DeeplabV3+: https://github.com/tensorflow/models/tree/master/research/deeplab

The algorithm was trained and tested on publicly available datasets. Therefore, the model's data loaders were extended to use publicly available datasets (A2D2) as well as the synthetic generated dataset from the project.

The algorithm is provided in the projects Bitbucket (previously gitlab) and made available to the project in software releases. The algorithm can be found here:

- DeeplabV3+: https://luxproject.luxoft.com/stash/projects/KIA/repos/tp1_ap1.3_semantic_segmentation_deeplabv3plus/browse

For each algorithm trained weights are provided in the projects data storage. The following table gives an overview on the available weights:

Algorithm	Dataset	Weights
DeeplabV3+	a2d2	https://kip.gpu-cluster.itwm.fraunhofer.de/minio/lfs-objects/TP1/AP1.3/SemanticSegmentationIntelZF/DeeplabV3+/a2d2/
DeeplabV3+	KIA-Tranche1 BIT-TS (Sequenz 1-24)	https://kip.gpu-cluster.itwm.fraunhofer.de/minio/lfs-objects/TP1/AP1.3/SemanticSegmentationIntelZF/DeeplabV3+/kia/
DeeplabV3+	KIA-Tranche2 BIT-TS (Sequenz 25-69)	https://kip.gpu-cluster.itwm.fraunhofer.de/minio/lfs-objects/TP1/AP1.3/SemanticSegmentationIntelZF/DeeplabV3+/kia_tranche2_bit-ts/
DeeplabV3+	KIA-Tranche3 BIT-TS	https://kip.gpu-cluster.itwm.fraunhofer.de/minio/lfs-objects/TP1/AP1.3/SemanticSegmentationIntelZF/DeeplabV3+/kia_tranche3_bit-ts/
DeeplabV3+	KIA-Tranche3 BIT-TS and KIA-Tranche4 BIT-TS	https://kip.gpu-cluster.itwm.fraunhofer.de/minio/lfs-objects/TP1/AP1.3/SemanticSegmentationIntelZF/DeeplabV3+/release3/
DeeplabV3+	KIA-Tranche7 MackeVision	https://kip.gpu-cluster.itwm.fraunhofer.de/minio/lfs-objects/TP1/AP1.3/SemanticSegmentationIntelZF/DeeplabV3+/release4/

Weights provided for KIA Tranche3 forward are without pretrained backbone, to prevent license issues. Furthermore the code has been adopted for Automated Smoke Tests according to the P2 release process.

Additionally, predictions on the test set are provided alongside a metadata. json file and a greyscale image with information on the prediction confidence, i.e. confidence of [0%,100%] is mapped to [0,255] greyscale (see image below)

Evaluation results on the KIA Tranche5 set were provided with the weights previously trained on KIA Tranche3 and KIA Tranche4.

3.1.2.2 Examples

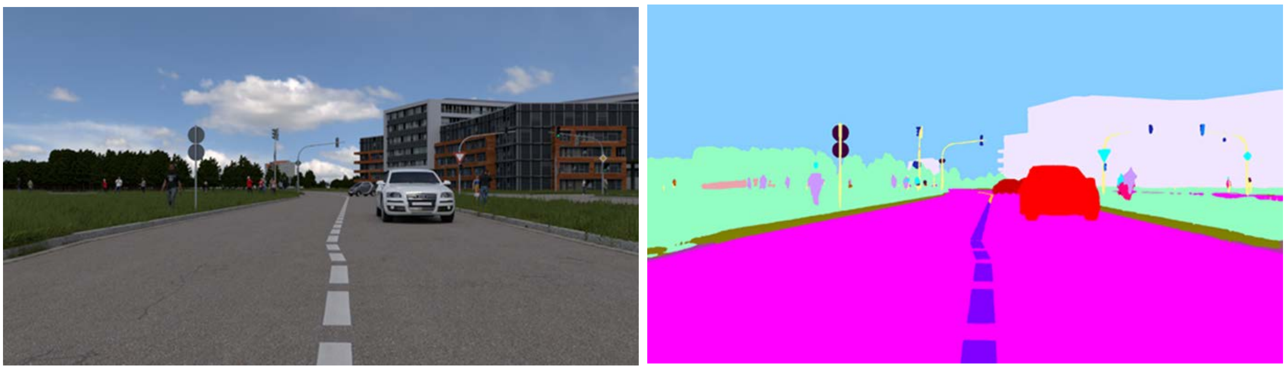


Figure 1: Synthetic image (left) and predicted pixel accurate semantic segmentation (right)

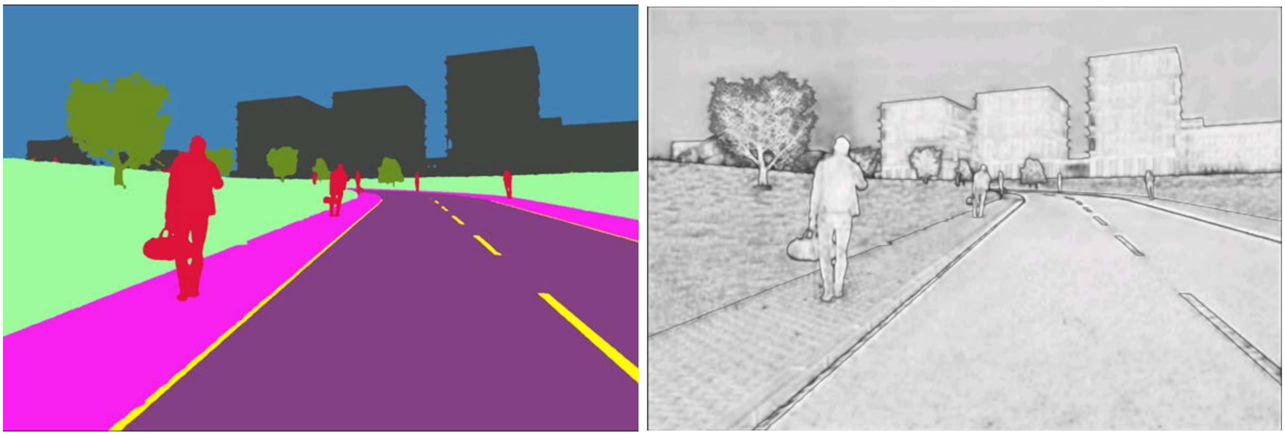


Figure 2: Prediction (left) and confidence score (right) on KIA Tranche3 and Tranche4

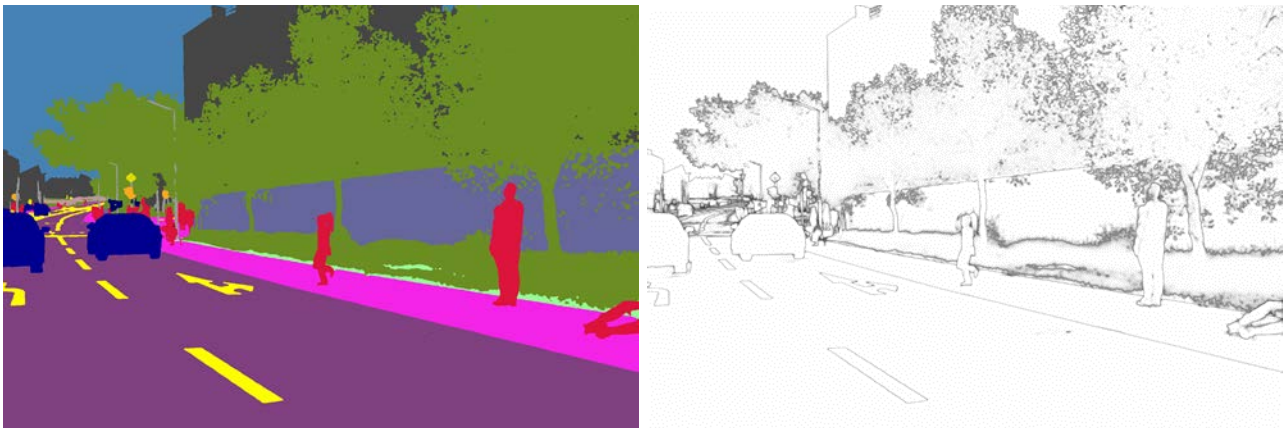


Figure 3: Prediction (left) and confidence score (right) on KIA Tranche7 MV

3.1.2.3 DeeplabV3

For the task of semantic segmentation, we find Deeplab v3 [1] to deliver state of the art performances. We train Deeplab v3 on the A2D2 dataset [2] as well as the project internal KIA dataset. Inference samples can be found in section Examples. The A2D2 dataset is a real-world dataset created by Audi and labelled with semantic segmentation classes. The data domain is within autonomous driving and the class set includes ADAS related classes such as pedestrians and cars. Additionally, we train Deeplab v3 on the KIA dataset. This dataset contains semantic segmentation labels for synthetic images. Both images and labels are generated automatically and synthetically. The data domain is within autonomous driving as well and classes are similar to those of the Cityscapes dataset, which share properties of the A2D2 dataset. Therefore, there is a class overlap of A2D2 and KIA.

3.1.3 Results

Deeplab v3 performs semantic segmentation of images with a deep convolutional neural network. It belongs to the family of fully convolutional networks, which transform images to labels of the same dimensional space as the input without the use of fully connected layers. A convolutional backbone layer stack (here: ResNet) processes the image. The last layers of the network incorporate dilated convolutions for a wider field of view of individual filters and thereby allow for higher classification performances.

Deeplab v3 is trained on the A2D2 dataset, which includes a class for 'pedestrian'. For this dataset, labels are converted from human readable RGB labels to categorical labels. When there are multiple label classes for the same category (instance separation), they are merged. Hyperparameters of Deeplab v3 are adapted to be suitable for this task. This includes adapting hyperparameters to the given hardware, dataset and training schedule. The code and dataset will be made available on:

https://gitlab.com/kia2/tp1/ap1.3/semantic_segmentation/deeplabv3 (deprecated)

https://luxproject.luxoft.com/stash/projects/KIA/repos/tp1_ap1.3_semantic_segmentation_deeplabv3/browse (current)

An inference on an A2D2 test set sample can be found in Grafik A. Inference on synthetic data from TP2 can be found in Grafik B.

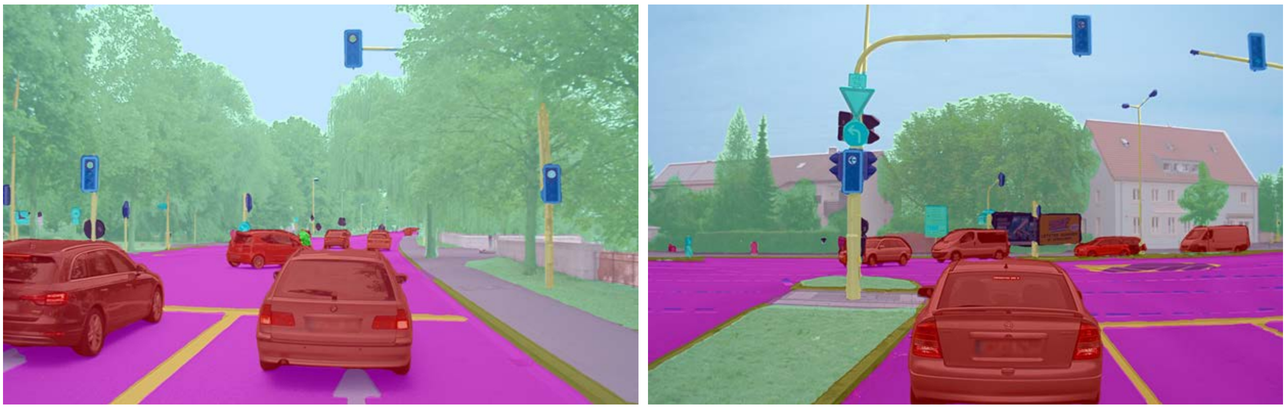
Weights for DeeplabV3 are made available for different intput resolutions / GPU resources. The Input-Output pair is downscaled between 8 times and the original resolution. Downscaling denotes the resizing of the image/label pair by a factor. In the original resolution, Deeplabv3 reaches a high accuracy of 57.55% mIOU on A2D2. Further details can be found in Gitlab/Bitbucket.

1	57.55%	https://kip.gpu-cluster.itwm.fraunhofer.de/minio/download/lfs-objects/TP1/AP1.3/SemanticSegmentationIntelZF/Deeplabv3/deeplab_resnet101_a2d2_v3_down_1_epoch100.pth
2	52.18%	https://kip.gpu-cluster.itwm.fraunhofer.de/minio/download/lfs-objects/TP1/AP1.3/SemanticSegmentationIntelZF/Deeplabv3/deeplab_resnet101_a2d2_v3_down_2_epoch200.pth

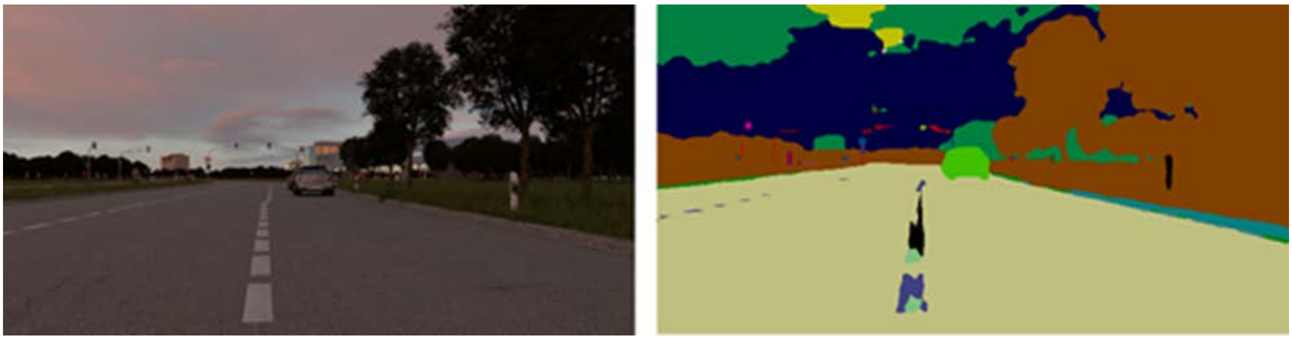
4	37.96%	https://kip.gpu-cluster.itwm.fraunhofer.de/minio/download/lfs-objects/TP1/AP1.3/SemanticSegmentationIntelZF/Deeplabv3/deeplab_resnet50_a2d2_v3_down_4_epoch100.pth
8	28.22%	https://kip.gpu-cluster.itwm.fraunhofer.de/minio/download/lfs-objects/TP1/AP1.3/SemanticSegmentationIntelZF/Deeplabv3/deeplab_resnet50_a2d2_v3_down_8_epoch100.pth

For the KIA dataset, the weights are first converted to a Cityscapes-readable format and the trained. This includes a converter for the KIA dataset and dataloader for the Deeplab network. The network is trained on Tranche 3 and reaches an accuracy of 81.57% on the respective test set. It is trained with a resnet50 backbone for 100 epochs.

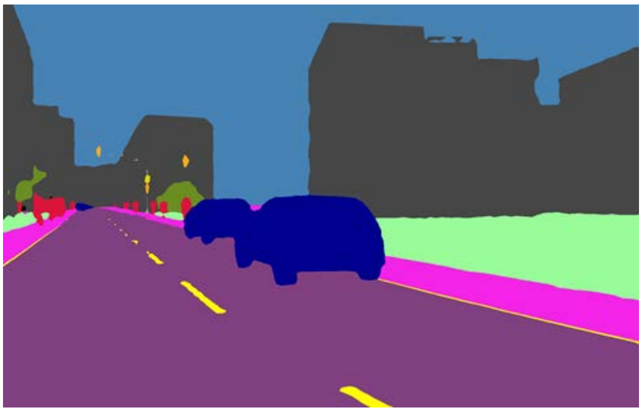
3.1.3.1 Examples



A: Deeplab trained on A2D2 / Evaluated on A2D2 Test (newest weights)



B: Deeplab trained on A2D2 / Evaluated on TP2 Synthetic Data (previous weights)



*C: Deeplab v3 trained on **KIA** Data.*

3.1.4 References:

[1] Chen, Liang-Chieh et al. “Rethinking Atrous Convolution for Semantic Image Segmentation.” ArXiv abs/1706.05587 (2017): n. pag. https://arxiv.org/abs/1706.05587

[2] https://www.audi-electronics-venture.com/aev/web/en/driving-dataset/dataset.html

3.2 E1.3.3c Final: nur projektintern für KI Absicherung verfügbar

3.3 E1.3.3d Final: nur projektintern für KI Absicherung verfügbar

3.4 E1.3.3e Final: nur projektintern für KI Absicherung verfügbar

3.5 E1.3.4 Final: nur projektintern für KI Absicherung verfügbar

3.6 E1.3.5 Final: nur projektintern für KI Absicherung verfügbar

4 AP1.4 Tiefendaten-Algorithmen

4.1 E1.4.1 Final: nur projektintern für KI Absicherung verfügbar

4.2 E1.4.3 Final: Fusion auf Regressionsebene (zur Veröffentlichung)

4.2.1 Formal Classification

Criteria	Classification according to VHB
Type of result	*Code*
Group/Cluster	
Type of content	*DL Model*
Classification level	*PU*

4.2.2 Description of the result

The goal of this work package is to implement an algorithm for 3D Box Detectors on lidar with late fusion with images of pedestrians.

4.2.3 Approach

For the task of 3D bounding box detection with late fusion of lidar and image data, we find F-ConvNet[1] to deliver state of the art performances. We evaluated F-ConvNet on several public datasets and the dataset from TP2 (The "KIA dataset") and perform inference on samples generated by TP2.

4.2.4 Result

We worked on KITTI Format Converters where we implement one for the A2D2 [2] and one for KIA dataset. This converted KITTI Format makes it possible that the datasets can be used without any model code adaptions.

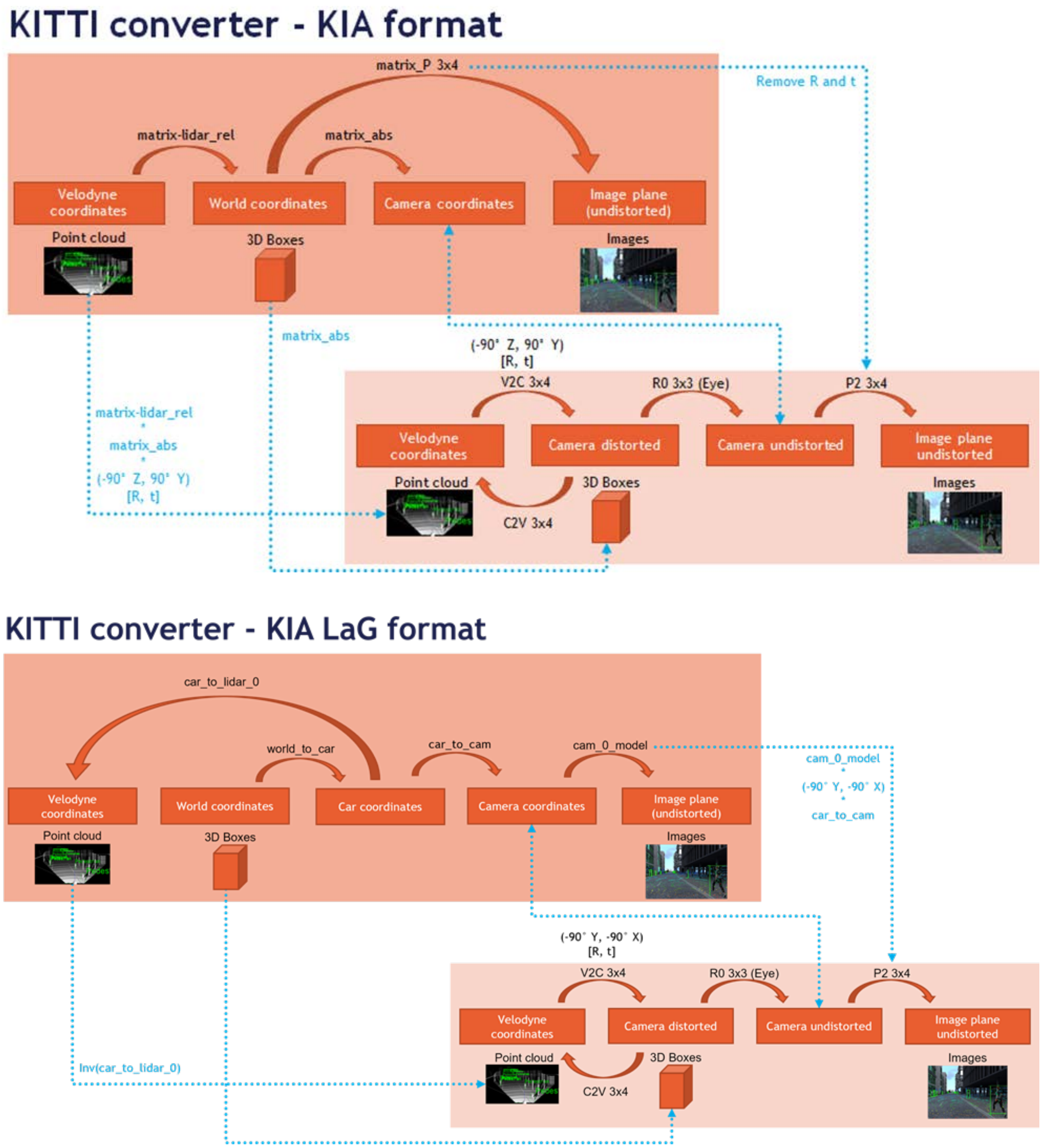


Figure 1: Illustration of converting from KIA to KITTI format

The illustration in Figure 1 clarifies the KITTI to KIA conversion process: Upper left KIA format, bottom right the KITTI format. In the KITTI Format, the 3D box center is at the bottom and the up-vector is the y-axis, whereby in KIA it is the center and the up-vector z-axis.

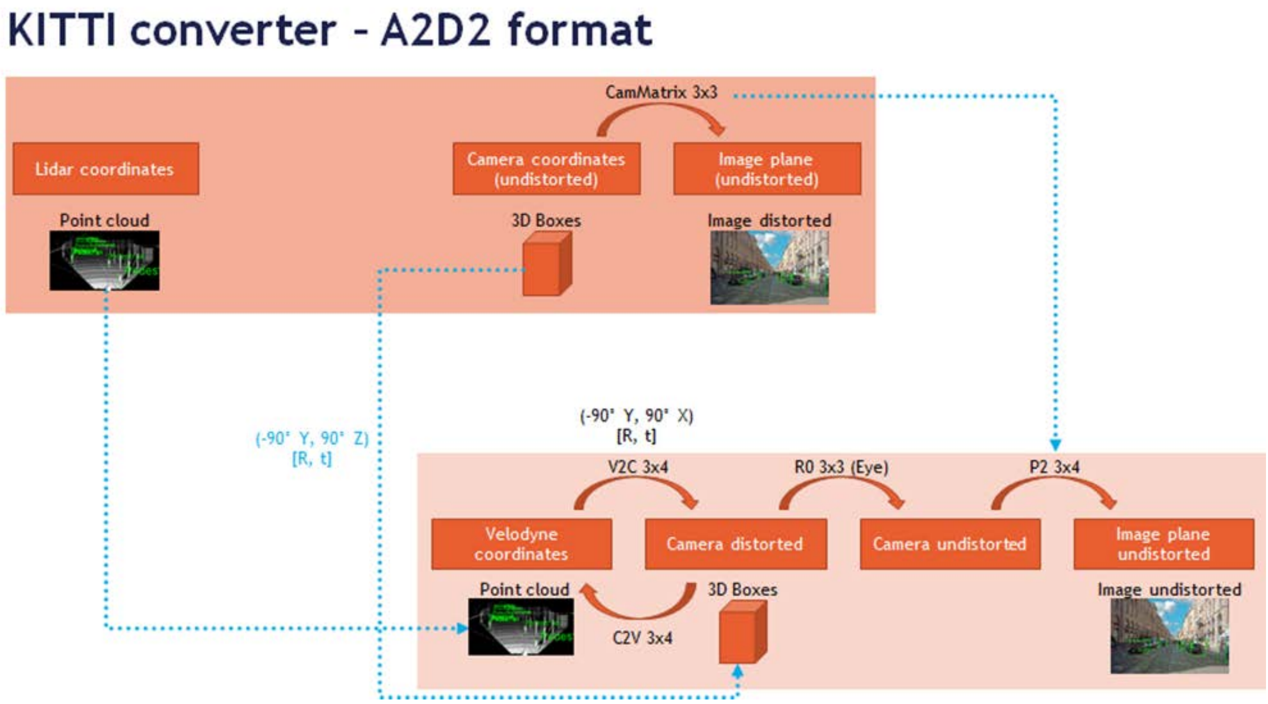


Figure 2: Illustration of converting from A2D2 to KITTI format

The illustration in Figure 2 clarifies the A2D2 to KITTI conversion process: Upper left A2D2 format, bottom right the KITTI format. In the KITTI Format the 3D box center is at the bottom and the up-vector is the y-axis, whereby in A2D2 it is the center and the up-vector z-axis. In A2D2, all coordinates are in the car world system which is not illustrated here.

A full evaluation on four datasets was made: KIA Tranche 3 Bit TS, A2D2 [2], KITTI [4] and NuScenes [5] which can be seen in the table below. For A2D2, an own data split was made because of the unavailability of an official one. We removed also all sequences from the A2D2 dataset with less than 34 pedestrians per sequence to improve the results. For NuScenes, a slightly modified version of the official KITTI Converter was used to fit our purpose. Additionally, samples in which the 2D box is not in the field of view were removed from the NuScenes data set. The KITTI and NuScenes dataset officially have only a validation set. The KIA dataset has a validation and a test set.

We also participated in Release 2*. The delivered software package does not pass the tests yet, because of GPU / CUDA issues on the test bench, which will be considered later from the release management process side. In order to deliver a functional and usable software (even if it does not pass the Release 2* tests as mentioned), we added all necessary instructions to the README. Further, we uploaded a trained model of the F-ConvNet [1] on the KIA Tranche 3 Bit-TS data set to the project's model repository.

In our last report, we reported issues related to the 3D bounding box annotations of the KIA data. Unfortunately, these problems could be solved yet, but because of the short period we are still on a new evaluation.

The following two tables show some information about the datasets.

Dataset	Training	Validation	Test
A2D2	4858	2002	1845
KIA T3	21803	5164	9458
KIA T3,4,5 LaG	32189	8741	19664
KITTI	3712	3769	
NuScenes(Key-frames)	16097	6019	

Table 1: an overview of the number of samples in the data set splits.

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

Table 2: difficulty levels according the widely used KITTI benchmarks.

Dataset			Difficulties	Easy		Moderate		Hard	
			Threshold	0.5	0.25	0.5	0.25	0.5	0.25
	Best Model Epoch	**Maximal Distance (meters)**	Split						
KITTI [4] (2D box proposals from the paper)	24	70	val	67.07	87.25	58.46	81.57	51.87	74.66
KITTI [4] (ground truth proposals)	24	100	val	70.56	90.27	**63.94**	**90.18**	**62.93**	**89.72**
KIA T3	34	100	val	**75.43**	90.22	58.47	72.22	58.47	72.22
			test	66.91	**90.35**	57.61	72.27	57.61	72.27
A2D2 [2]	25	100	val	60.10	87.15	61.30	79.23	54.40	78.91
			test	61.18	87.42	61.52	79.20	54.44	78.72
NuScenes [5]	23	100	val	43.13	66.14	41.80	65.53	41.80	65.53
KIA T3,4,5	31	100	val	56.63	72.24	47.75	62.83	47.75	62.83
			test	44.94	62.96	36.09	53.41	36.09	53.41

Table 3: Evaluation of the Frustum-Conv-Net.

We trained each model 50 epochs and used the best model epoch for evaluation. The training used here is without the refinement stage (please see the paper or our last report for more information). A sample is counted as true positive, if the overlap is larger than the given threshold shown in the table header. The explanation of the difficulties can be found in Table2.

In the table above, the best results could be achieved on the KITTI data set with ground truth annotations with the proposed data set split of the paper. The maximal distance (third column) plays an important role in the results quality. We used 100 meters as maximum distance for all data sets which leads to better results compared to the 70 meters. The results of the KIA data set look almost as good as the KITTI results, which could be explained by the testing only on simulation data which has the same training distribution. The A2D2 has quite sparse 3D point clouds, which is related to the scan pattern of the lidar sensors. This can be seen in the inference image below. The bad results obtained for the NuScenes data set could be also due to the sparse 3D point clouds and the high data variation.

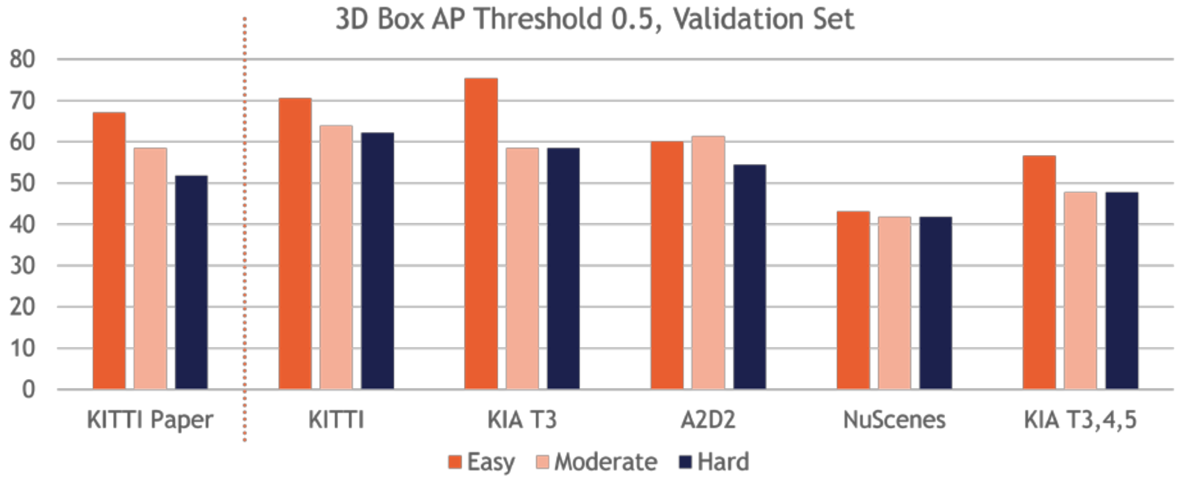


Figure 3: Evaluation on different datasets with different difficulty levels

The Illustration in Figure 3 of the validation sets in the table above. On the left-hand side of the dashed border you can see the first row in Table 3 (70 m maximal distance, 2D box proposals from the paper). On the right-hand side, we have 100 meters maximal distance and we use 2D box ground truth proposals.

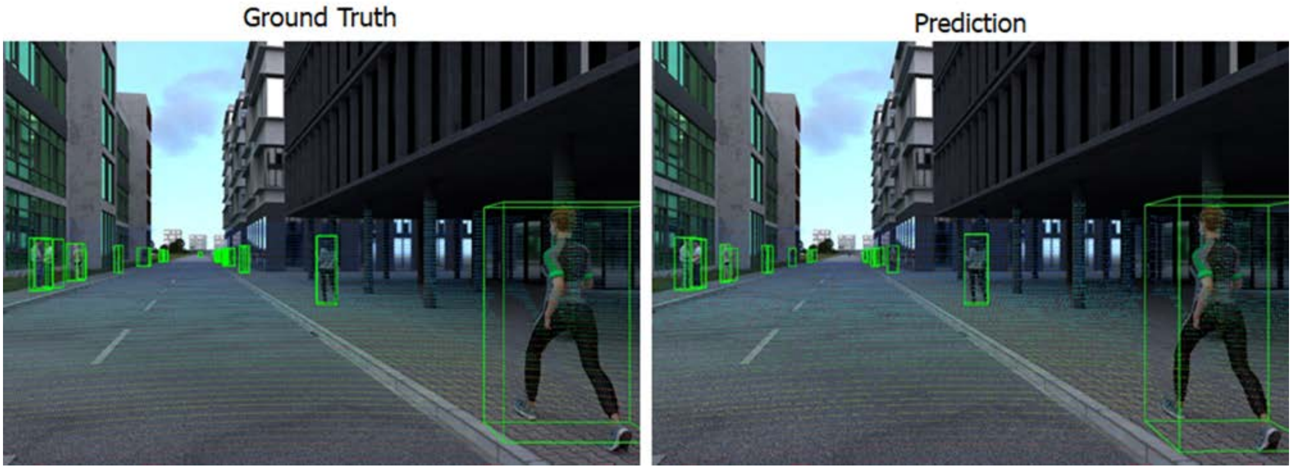


Figure 4: Comparison between ground truth and prediction for KIA Tranche 3.

Figure 4 shows an inference sample of Tranche 3 BIT TS (test set sequence 122). Sometimes the prediction bounding boxes are better than the ground truth bounding boxes due to errors in the ground truth.

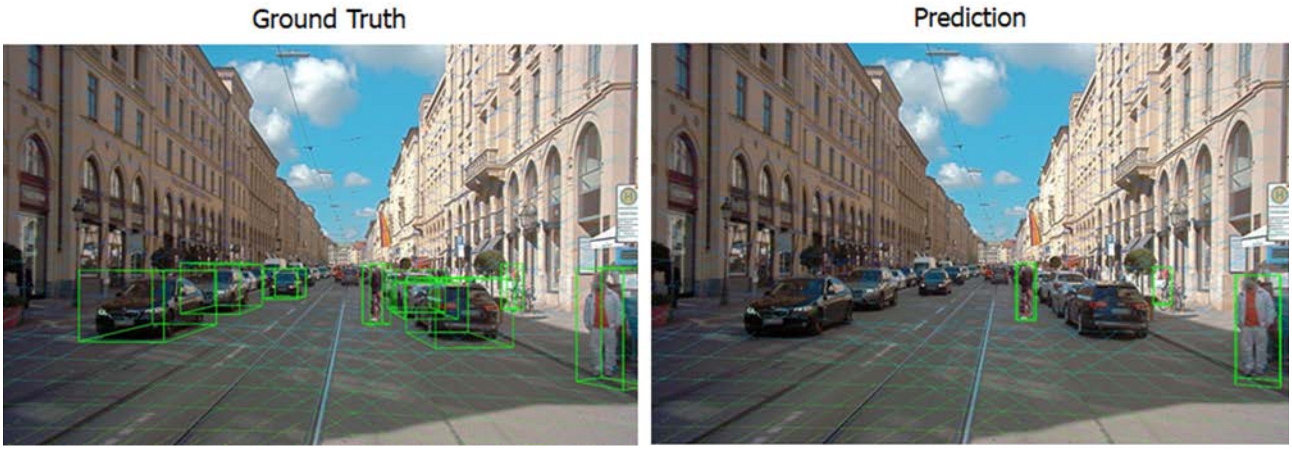


Figure 5: Comparison between ground truth and prediction for A2D2.

Figure 5 shows an inference sample of A2D2 - The 3D scan pattern is quite sparse as can be seen in the pictures.

References

[1] Wang et al., "Frustum ConvNet: Sliding Frustums to Aggregate Local Point-Wise Features for Amodal 3D Object Detection", IROS 2019

[2] Geyer, J., Kassahun, Y., Mahmudi, M., Ricou, X., Durgesh, R., Chung, A. S., ... & Fernandez, T. (2020). A2d2: Audi autonomous driving dataset. *arXiv preprint arXiv:2004.06320*.

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

[4] Andreas Geiger, Philip Lenz, & Raquel Urtasun (2012). Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In Conference on Computer Vision and Pattern Recognition (CVPR).

[5] Caesar, Holger, et al. "nuscenes: A multimodal dataset for autonomous driving." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020.

4.3 E1.4.4 Final: Single Modalität Lidar (zur Veröffentlichung

4.3.1 Formal Classification

Criteria	Classification according to VHB
Type of result	Code
Group/Cluster	Implementation of methods
Type of content	DL-Model
Classification level	PU

4.3.2 Introduction and motivation

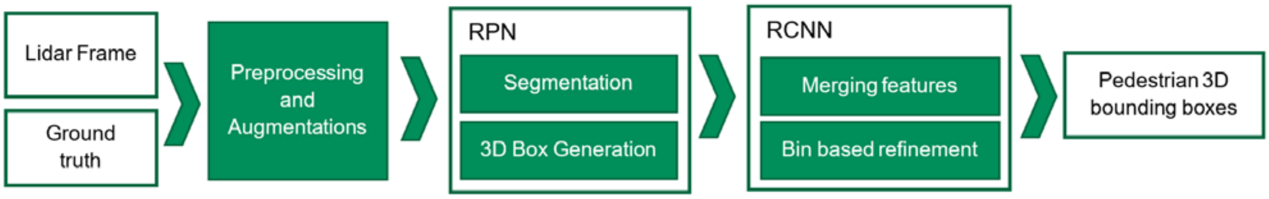
In this working package, we look at the problem of pedestrian detection from Lidar only point cloud. This does not include any fusion with images or other sensors and relies only on 3D sparse data. With the advances in deep learning, scientists have developed models that can work on set and unordered data such as points. Working on a large scale and space point cloud still is a challenge and in this project, we studied different models and techniques to sample, process, and learn pedestrians and objects from large-scale lidar clouds. Through the project, we looked into Birds-Eye-View (BEV) methods, point-based and voxel-based methods and finally come up with a point-voxel-based approach to detect pedestrians.

4.3.3 Approach

Based on an extensive literature review on 3D object detection which incorporates point cloud data, we see different categories of methods. Rather than using raw point cloud processing or voxelization approaches, we aimed to use RCNN-based approaches that have been successful in 2D Object detection as well. We utilized PointPillars[5] PointRCNN [1] and PV-RCNN [3] methods which have been the state-of-the-art approach on LiDAR-only object detection.

The method and the implemented code on the project repository are based on PointNet++, which incorporates hierarchical point cloud segmentation. Extracted deep features can be used to predict object bounding boxes, which here are restricted to the single class pedestrian boxes. For training at this stage ground truth labels and annotations are needed for augmentation and foreground-background segmentation. The RPN stage produces bounding box proposals for the next RCNN stage.

Each bounding box around the pedestrian instance is then taken into its canonical coordinates, and the features are merged with segmentation features and passed to a point cloud encoder to form output values. Bins of bounding boxes and their position and rotation are predicted using a combination of classification and regression loss functions alongside a confidence score. The output predictions are then moved to the world coordinates reference frame for validation and visualization.



Due to data constraints of the previous AI-Function based on PointRCNN [1] such as lack of ground plane annotations, offline data preprocessing, and two-stage training scenario, PointRCNN does not provide generalizability to new datasets. Therefore we integrated Point-Voxel based Feature Set Abstraction, PVRCNN [3] with less training complexities, more flexibility, and improved metric results on publically available datasets. We used a combination of voxel-based sparse convolution and set abstraction inspired by PointNet networks [2]. The method first processes the point cloud into voxel and sparse convolutions to generate regions of interest. Following the proposal generation, we use keypoints and voxel set abstraction to summarize the features and refine the proposals.

We continued our work based on the PVRCNN [3] to improve the performance by changing the way of extracting the features from the LIDAR point cloud using graph convolutional neural networks and transformers to use the attention to focus on the most important features and balance the inhomogeneous point sampling of the LIDAR scanner. Furthermore, we noticed that in some cases the predicted bounding boxes are off or missing completely for more difficult objects, such as far away objects with low LIDAR coverage or heavily occluded objects. Some recent approaches [4] try to complete the sparse LIDAR scans prior to running an object detection algorithm to increase the density of the point cloud, while this can help with object detection and estimating their extent it can also lead to false detections as potentially erroneous data is imputed. We, therefore, worked on a different approach to extract all the available contextual information from the LIDAR point cloud also over larger distances. Starting with the proposals generated from the PVRCNN [3] pipeline we aim to refine the predictions by connecting them with each other and with other parts of the environment using graph neural networks. We incorporate the pose information and relationships between different objects and use message passing to learn scene level priors about the distribution of objects with respect to each other and with the environment e.g. the road layout. This approach shows some improvement on the publically available datasets, but it does not transfer well to the KIA dataset. This is due to the different focus and distribution of object classes, while KITTI includes a majority of vehicles and only some pedestrians, the KIA dataset focuses on pedestrians and only includes a few vehicles.

Towards the end of the project, we worked towards a publication of the results we obtained during our latest experiments on using Graph Convolutional Networks for Object Detection in raw LiDAR point clouds. As we noticed some differences between the detection of larger object classes such as cars and smaller objects such as pedestrians, we investigated the limitations with respect to sensor coverage and sampling resolution as well as to object complexity.

4.3.4 Results and Evaluation

In the official releases, we provide python code and scripts required to train and validate 3D detection neural networks. Given the provided dataset and its required annotations, the scripts first perform preprocessing and augmentations to train. The rest of the code consists of network modules including third-party dependencies (pointnet++ and roipooling), training stage one, stage two, and validation of the end-to-end pipeline.

Using a two-stage approach based on separate training sessions, we manage to detect and fit bounding boxes more accurately than the single-stage approaches. We have modified the official repository to detect pedestrian classes and measure metrics and visualize the predicted bounding boxes with poses. The output metric for our results is chosen as Intersection over Union (IoU). We calculated so far, the results based on publicly available benchmarks.

Table 1: Results of the validation set

IoU	**Easy**	**Medium**	**Hard**
0.5	64,3349	58,0765	51,2030
0.25	78,1504	71,6095	65,7971

4.3.5 Conclusion

In this working package, we focused on lidar and point cloud only pedestrian detection using different backbones. By incorporating a two-stage approach based on RCNN methods with separate training sessions, we manage to detect and fit bounding boxes more accurately than the single-stage approaches. Compared to point only and BEV methods, hybrid voxel and point-based methods prove more accurate results. Moreover, by leveraging contextual information using scene graphs we manage to improve our detection metrics.

4.3.5.1 References

[1] Shi, Shaoshuai, Xiaogang Wang, and Hongsheng Li. **"Pointrcnn: 3d object proposal generation and detection from point cloud."** *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019.

[2] Qi, Charles Ruizhongtai, et al. **"Pointnet++: Deep hierarchical feature learning on point sets in a metric space."** *Advances in neural information processing systems*. 2017.

[3] Shi, Shaoshuai, et al. "**Pv-rcnn: Point-voxel feature set abstraction for 3d object detection."** *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.

[4] Zhang, Y., Huang, D., & Wang, Y., **"PC-RGNN: Point Cloud Completion and Graph Neural Network for 3D Object Detection"**, *Proceedings of the AAAI Conference on Artificial Intelligence.* 2021.

[5] Lang, Alex H., et al. **"Pointpillars: Fast encoders for object detection from point clouds."** Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

4.4 E1.4.5 Final: nur projektintern für KI Absicherung verfügbar

4.5 E1.4.6 Final: nur projektintern für KI Absicherung verfügbar

5 AP1.5 - Posenschätzungs-Algorithmen

5.1 E1.5.1 Final: Angaben zum Stand der Wissenschaft (zur Veröffentlichung)

5.1.1 Formal Classification

Criteria	Classification according to VHB
Type of result	*Document*
Group/Cluster	
Type of content	*State of the art*
Classification level	*PU*

5.1.2 Description of the result

The state-of-the-art research has been done by four partners with different focus. The focus has been on 3D pose estimation from RGB and depth-data, 2D-/3D-pose estimation from RGB-only for self supervised learning, 2D-/3D-pose estimation from RGB-only for back-projecting methods and fusion of pose estimation and person detection.

Literature Review with focus towards 3D Pose Estimation from RGB and Depth-Data

Before 2014 classical approaches for pose estimation with bio-mechanical constraints were used. Deep Pose [1] extended Alexnet with regressors for joints and in an iterative process improved precision with a cascade of regressors. [3] introduced a multi-resolution architecture and heatmaps for prediction joints, which are the foundation of many approaches such as [4]. CPM [4], [5] further improved the quality of predictions. Stacked hourglass networks [6] consist of so called hourglass modules, which abstract while reducing and restoring the resolution. [7] on the other hand achieves state-of-the-art performance with a simple architecture, using ResNet and deconvolution layers. In contrast to previous approaches HRNet [8] keeps the resolution, leading to better results on COCO [9]. COCO consists of RGB images and 2D Annotations, like Penn Action [10], LSP Dataset [11] and MPII HP [12]. For 3D pose estimation from RGB images there are various approaches. [13] was the first approach modeling 3d joint estimation as a detection problem. [14] predicted first the position of the joints in 2d and then transformed them to 3d with a probabilistic model, a big database of 3d poses is matched. One advantage is the possibility to use huge databases of 3d poses without the need for RGB images for every pose. [15] extended this idea by a geometric loss function to allow training on 3D and 2D data for 3D pose estimation. [16] and [17] improve 3D pose estimation on RGB images further. [18] explores possibilities to reduce the amount of required training data, by first training an encoder-decoder network, which learns to predict a different view (e.g. from front view to side view). For this step no annotations are required, just RGB images of the same scene and subject from different views. Using the latent representation, a neural network learns to predict the 3D pose. This network can have very few learnable parameters and therefor allows for reduced number of training samples. LCR-Net [19] predicts a region proposal and a pose proposal, which is refined in a second stage. Further approaches for 3D Pose Estimation from RGB data are [20], [21], [22] and [23]. A common evaluation metric between the approaches is “mean per joint position error” (MPJPE) on datasets like 3DPW [24], MPI-INF-3DH [25], Human3.6M [26] and HumanEva

[27]. These datasets contain RGB images and 3D annotations. The only publicly available dataset providing LiDAR and RGB is PedX [28]. However, PedX did not publish 3D pose annotations. This renders 3d pose estimation technically impossible on PedX. Bio-LSTM [29] is the only currently available approach on PedX. It focuses on precise pose estimation from multiple unprecise temporal consecutive predictions regarding biological constraints.

As of today the only approach for Fusion of RGB and LiDAR data in Human Pose Estimation is our HPERL approach [36]. In 3D Object Detection fusion has been researched by various approaches and served as a foundation for our HPERL publication. The fusion approaches can be categorized into 4 groups: Early Fusion, Sequential Fusion, Late Fusion and Slow Fusion. In Early Fusion either the LiDAR data is converted to a depth map and concatenated to the image in an RGB-D Image or a pointcloud colorized by the rgb image. As Wang et al. [37] elaborates, the gap in performance between RGB only approaches and those which use LiDAR can be partially mitigated by choosing the right representation. With PseudoLiDAR [37] they show, that converting a depth map to a pointcloud and then detecting on that has superior performance. Thus also making early Fusion using RGB-D images a suboptimal choice consequently. In Sequential Fusion a first stage only uses one sensor and the second stage uses the extracted information plus the other sensor. PointPainting [38] first computes the semantic segmentation on the RGB image and then paints the points in the 3d pointcloud using that segmentation. Then a lidar based detection is applied on the painted pointcloud. It shows good results, but is limited to the density of the pointcloud. Late Fusion approaches use both sensors to create predictions and fuse them at a late stage in the network. An example would be AVOD [39] which computes the features on RGB-Image and BEV-Pointcloud separately and only fuses features for anchorcrops. In Slow Fusion the features of different views are fused within the feature encoding enabling a dense information exchange between views. An example would be ContFuse [40]. In End-To-End driving TransFuser [41] published on CVPR 2021 has introduced a slow fusion using transformers. TransFuser lacks the spatial output grid required for detection and human pose estimation though.

Literature Review with Focus towards self-supervised 2D-/3D-Pose Estimation from RGB-only

Unsupervised methods for Human Pose Estimation can be seen as a specialization of unsupervised landmark recognition approaches for the for the human body. Landmark recognition aims at finding discriminative points on a possibly large number of objects and has been successfully applied to instances such as human faces in an unsupervised manner [30], [31], [32], [33]. Nearly all of these methods make use of invariance or equivariance (see [34] for the idea of equivariance) constraints under different types of artificial image transformations and try to reconstruct a given original image as a surrogate task [31], [33] as this enforces model to be able to explain the object depicted in the image. In addition, Lorenz et al. [33] learn disentangled representations of an object's shape and appearance, what helps to find semantically meaningful parts and from these to regress meaningful landmark locations [33]. This method has been successfully applied to unsupervised landmark recognition of the human body which is a semantically more complex object category than faces.

An orthogonal approach compared to the aforementioned is to learn a network to capture similarities in object. [35] applies an optimization based clustering and batching algorithm which enables a neural network to learn a feature space that captures similarities of human body posture in an unsupervised manner. The human pose of a person in a given query image is then estimated by using a database of annotated images and finding the query's nearest neighbor from this database in the learned feature space.

Literature review with focus towards 2D-/3D-pose estimation from a single RGB image

Based on the availability of datasets with groundtruth annotation and to simplify the problem many approaches for 3d human pose estimation from a single RGB image focused on scenes with only a single person in them [42, 43]. While others specifically target scenes with multiple person, trying to solve the ambiguity in assigning limbs to persons [44, 45, 46, 47, 48].

Some approaches assume the 2d pose has already been estimated [45, 49, 50], while others try to estimate the 3d pose from the image [43, 48].

Ramakrishna et. al [49] used an optimization method to estimate the 3D pose from a given 2d pose leveraging anthropometric priors like the proportions of different bones to each other. While Martinez et. al [50] trained a neural network to predict the 3d pose from a given 2d pose, enabling the prediction of 3d pose even under stronger self-occlusion. Mehta et al. [45] in estimate relative 3d offsets between the joints based on the image and leverage them in addition to the 2d pose as input for a neural network to lift the pose to 3d.

For the case where the 2d pose is unknown, some works directly estimate the 3d pose from the image [48], while others break down the process into intermediate steps [42, 45], by first trying to detect the visible joints (and bones) of the human in the image and then assemble the 2d pose based on them. While this approach provides information about which joints are visible, the estimated 2d pose is also limited to them. Some works therefore estimate a completed pose from the partial pose of visible joints [45].

In the case of multiple persons close to each other in an image, the assignments from joints to persons can be challenging, therefore Cao et al. [44] proposed to additionally predict affinity fields along the human bones to connect the joints in the image plane, leading to a much more stable matching process. Kreiss et al. [51] extended the idea to a different representation of affinity fields, not only encoding the direction of the bone, but also its start and end points.

While most works focus on estimating the relative 3d pose of the humans in the image, which is defined by the 3d offset of all joints with respect to a root joint [19, 45], some works in addition estimate the distance to the root joint to recover the absolute 3d poses [48, 52].

As the problem of absolute distance estimation from a single RGB image is generally ill-posed, there are different assumptions being made. Moon et al. [48] assume a standing person with a constant body height to estimate the distance to the person. As a consequence the methods performance degrades, if the person is not standing straight or the height of the person differs significantly from the average height.

Literature

[1] A. Toshev and C. Szegedy, “DeepPose: Human pose estimation via deep neural networks,” IEEE Conf. Comput. Vis. Pattern Recognit., 2014.

[3] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, “Efficient object localization using Convolutional Networks,” IEEE Conf. Comput. Vis. Pattern Recognit., 2015.

[4] S. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, “Convolutional Pose Machines,” 2016.

[5] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik, “Human pose estimation with iterative error feedback,” IEEE Conf. Comput. Vis. Pattern Recognit., 2016.

[6] A. Newell, K. Yang, and J. Deng, “Stacked Hourglass Networks for Human Pose Estimation.”

[7] B. Xiao, H. Wu, and Y. Wei, “Simple baselines for human pose estimation and tracking,” Lect. Notes Comput. Sci., 2018.

[8] K. Sun, B. Xiao, D. Liu, and J. Wang, “Deep High-Resolution Representation Learning for Human Pose Estimation,” 2019.

[9] T. Y. Lin et al., “Microsoft COCO: Common objects in context,” in European Conference on Computer Vision, 2014.

[10] W. Zhang, “From Actemes to Action : A Strongly-supervised Representation for Detailed Action Understanding,” pp. 2248–2255, 2013.

[11] S. Johnson and M. Everingham, “Clustered pose and nonlinear appearance models for human pose estimation,” Br. Mach. Vis. Conf., 2010.

[12] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, “2D Human Pose Estimation: New Benchmark and State of the Art Analysis,” IEEE Conf. Comput. Vis. Pattern Recognit., 2014.

[13] S. Li and A. B. Chan, “3D human pose estimation from monocular images with deep convolutional neural network,” Lect. Notes Comput. Sci., 2015.

[14] C. H. Chen and D. Ramanan, “3D human pose estimation = 2D pose estimation + matching,” IEEE Conf. Comput. Vis. Pattern Recognit., 2017.

[15] X. Zhou, Q. Huang, X. Sun, X. Xue, and Y. Wei, “Towards 3D Human Pose Estimation in the Wild: A Weakly-Supervised Approach,” IEEE Int. Conf. Comput. Vis., 2017.

[16] J. Martinez, R. Hossain, J. Romero, and J. J. Little, “A Simple Yet Effective Baseline for 3d Human Pose Estimation,” IEEE Int. Conf. Comput. Vis., 2017.

[17] X. Sun, B. Xiao, F. Wei, S. Liang, and Y. Wei, “Integral human pose regression,” Lect. Notes Comput. Sci., 2018.

[18] H. Rhodin, M. Salzmann, and P. Fua, “Unsupervised geometry-aware representation for 3D human pose estimation,” Lect. Notes Comput. Sci., 2018.

[19] P. Weinzaepfel and C. Schmid, “LCR-Net : Localization-Classification-Regression for Human Pose,” IEEE Conf. Comput. Vis. Pattern Recognit., 2017.

[20] B. Tekin, I. Katircioglu, M. Salzmann, V. Lepetit, and P. Fua, “Structured prediction of 3D human pose with deep neural networks,” Br. Mach. Vis. Conf., 2016.

[21] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, “Coarse-to-fine volumetric prediction for single-image 3D human pose,” IEEE Conf. Comput. Vis. Pattern Recognit., 2017.

[22] D. Mehta et al., “VNect : Real-time 3D Human Pose Estimation with a Single RGB Camera,” ACM Trans. Graph., 2017.

[23] R. A. Güler, N. Neverova, and I. Kokkinos, “DensePose: Dense Human Pose Estimation In The Wild,” IEEE Conf. Comput. Vis. Pattern Recognit., 2018.

[24] T. von Marcard, R. Henschel, M. J. Black, B. Rosenhahn, and G. Pons-Moll, “Recovering accurate 3D human pose in the wild using IMUs and a moving camera,” Eur. Conf. Comput. Vis., 2018.

[25] D. Mehta et al., “Monocular 3D Human Pose Estimation In The Wild Using Improved CNN Supervision,” IEEE Int. Conf. 3D Vis., 2017.

[26] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, “Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments,” IEEE Trans. Pattern Anal. Mach. Intell., 2013.

[27] “HumanEva: Synchronized Video and Motion Capture Dataset for Evaluation of Articulated Human Motion Leonid Sigal and Michael J. Black,” no. September, 2006.

[28] W. Kim et al., “PedX: Benchmark Dataset for Metric 3-D Pose Estimation of Pedestrians in Complex Urban Intersections,” IEEE Robot. Autom. Lett., 2019.

[29] X. Du, R. Vasudevan, and M. Johnson-Roberson, “Bio-LSTM: A biomechanically inspired recurrent neural network for 3-d pedestrian pose and gait prediction,” IEEE Robot. Autom. Lett., 2019.

[30] Thewlis, James, Hakan Bilen, and Andrea Vedaldi. "Unsupervised learning of object landmarks by factorized spatial embeddings." Proceedings of the IEEE International Conference on Computer Vision. 2017.

[31] Y. Zhang, Y. Guo, Y. Jin, Y. Luo, Z. He, and H. Lee. Un-supervised discovery of object landmarks as structural representations. In CVPR, 2018.

[32] T. Jakab, A. Gupta, H. Bilen, and A. Vedaldi. Conditional image generation for learning the structure of visual objects. NIPS, 2018.

[33] Lorenz, D, Bereska, L, Milbich, T and Ommer, B (2019). Unsupervised Part-Based Disentangling of Object Shape and Appearance. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (Oral + Best paper finalist: top 45 / 5160 submissions)*

[34] K. Lenc and A. Vedaldi. Learning covariant feature detectors. In ECCV Workshops, 2016.

[35] Bautista, Miguel & Sanakoyeu, Artsiom & Sutter, Ekaterina & Ommer, Björn. (2016). CliqueCNN: Deep Unsupervised Exemplar Learning.

[36] Michael Fürst, Shriya T. P. Gupta, René Schuster, Oliver Wasenmüller, Didier Stricker, "HPERL: 3D Human Pose Estimation from RGB and LiDAR", IEEE Int. Conf. on Pattern Regonit., 2021.

[37] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Killian Q. Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[38] Sourabh Vora, Alex H. Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[39] Ku, Jason, et al. "Joint 3d proposal generation and object detection from view aggregation." *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[40] Shenlong Wang, Simon Suo, Wei-Chu Ma, Andrei Pokrovsky, and Raquel Urtasun. Deep parametric continuous convolutional neural networks. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018

[41] Prakash, Aditya, Kashyap Chitta, and Andreas Geiger. "Multi-Modal Fusion Transformer for End-to-End Autonomous Driving." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* 2021.

[42] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G. Derpanis, Kostas Daniilidis, Coarse-to-Fine Volumetric Prediction for Single-Image 3D Human Pose, CVPR 2017

[43] Nibali, A., He, Z., Morgan, S., & Prendergast, L., 3d human pose estimation with 2d marginal heatmaps, WACV 2019

[44] Z. Cao, G. Hidalgo, T. Simon, S. -E. Wei and Y. Sheikh, OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields, TPAMI 2019

[45] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Mohamed Elgharib, Pascal Fua, Hans-Peter Seidel, Helge Rhodin, Gerard Pons-Moll, Christian Theobalt, XNect: Real-time Multi-person 3D Human Pose Estimation with a Single RGB Camera, CVPR 2019

[46] Li, J., Wang, C., Zhu, H., Mao, Y., Fang, H. S., & Lu, C., Crowdpose: Efficient crowded scenes pose estimation and a new benchmark, CVPR 2019

[47] Abdallah Benzine, Florian Chabot, Bertrand Luvison, Quoc Cuong Pham, Catherine Achard, PandaNet : Anchor-Based Single-Shot Multi-Person 3D Pose Estimation, CVPR 2020

[48] Gyeongsik Moon, Ju Yong Chang, Kyoung Mu Lee, Camera Distance-aware Top-down Approach for 3D Multi-person Pose Estimation from a Single RGB Image, CVPR 2019

[49] Ramakrishna, V., Kanade, T., & Sheikh, Y., Reconstructing 3d human pose from 2d image landmarks, ECCV 2012

[50] J. Martinez, R. Hossain, J. Romero, and J. J. Little, A Simple yet Effective Baseline for 3d Human Pose Estimation, CVPR 2017

[51] Kreiss, S., Bertoni, L., & Alahi, A., Pifpaf: Composite fields for human pose estimation, CVPR 2019

[52] Wang, C., Li, J., Liu, W., Qian, C., & Lu, C., Hmor: Hierarchical multi-person ordinal relations for monocular multi-person 3d pose estimation., ECCV 2020

5.2 E1.5.2 Final: Algorithmus zur 2D-3D-Posenschätzung aus reinen RGB-Daten (zur Veröffentlichung)

5.2.1 Formal Classification

Criteria	Classification according to VHB
Type of result	*Code*
Group/Cluster	
Type of content	*DL Model*
Classification level	*PU*

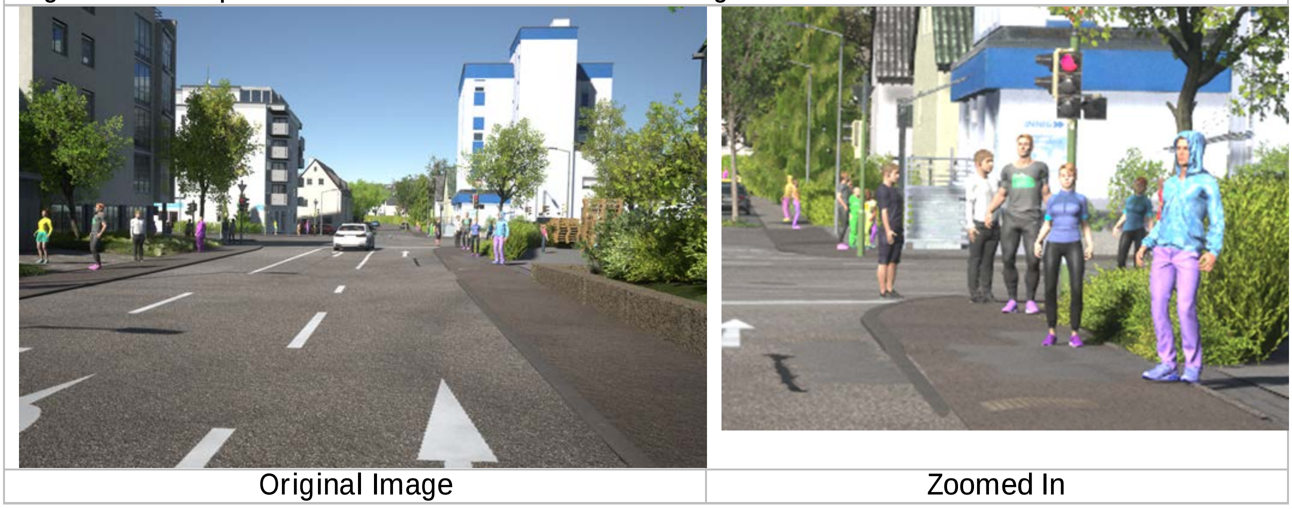
5.2.2 Description of the result

5.2.2.1 Motivation

The aim of this deliverable is the development and evaluation of a deep learning-based approach to estimate the 3D human pose from a single RGB image as a baseline implementation to be used during the project. We aim to compare different approaches with respect to their performance and reliability in autonomous driving scenarios based on the dataset developed during the project.

Compared to other sensors RGB cameras provide a high resolution signal of the environment, are widely available in mobile platforms and come at a comparably low cost. However the autonomous driving scenario poses some new challenges for existing methods for 3d human pose estimation. Due to the large field of view of the front facing camera the scale difference between persons standing near the car and the ones standing far away can be very large (see Fig. 1), therefore a tradeoff between including enough context of the person, allowing for a sufficiently high image resolution for body part detection and at the same time keeping the model computational and memory efficient is challenging. Another challenge is posed by the spatial distribution of persons in the image, in urban areas pedestrians tend to clutter and occur in smaller or larger crowds, here it is difficult to find the correct number of persons as representations like a 2d bounding box might be very similar for different persons and care has to be taken to not filter out these cases during non-maximum suppression. In cases of crowds with strong oclusions it is sometimes difficult to assign the body parts to the correct person, here different representations such as part affinity fields between the joints or instance masks of a person can help, but might also lead to wrong predictions. Finally the problem of absolute 3d pose estimation from a single RGB image itself is ill-posed as the 3d information is lost during the projection of the scene onto the image plane, without a known reference in the real world such as the actual height or bone lengths of the persons only context information from the scene itself can be leveraged, which is difficult for existing methods as they only focus on the nearby area around the person. It is also not directly possible to estimate the full skeleton of a person as multiple body parts can be occluded by the person itself or by other objects in the scene, therefore some approaches try to estimate the pose of the remaining body parts based on the ones that have been detected, for which there usually exists multiple plausible solutions.

Figure 1: Example scene from the autonomous driving context



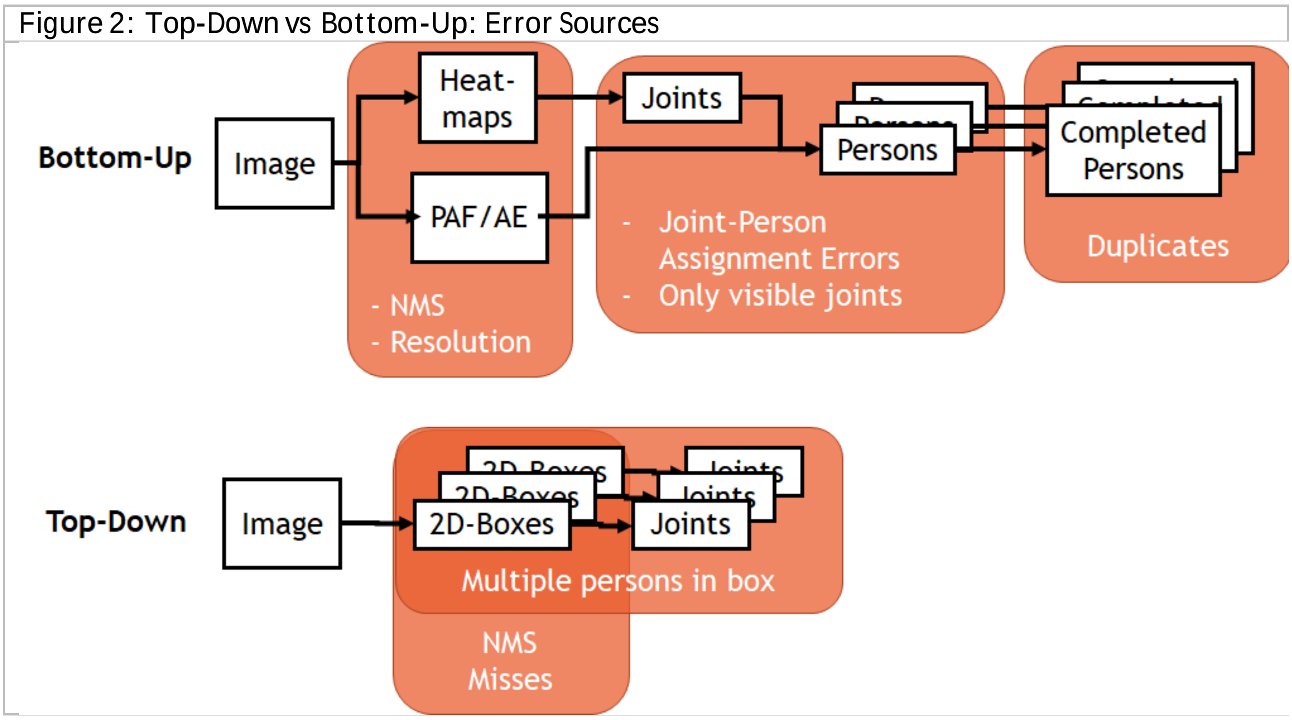
Original Image

Zoomed In

5.2.2.2 Approach

5.2.2.2.1 Baselines

Figure 2: Top-Down vs Bottom-Up: Error Sources



Existing approaches for multi 2d/3d human pose estimation can be split into two groups bottom-up and top-down methods, see also Fig. 2. While bottom-up approaches start by detecting individual body parts in the image and continue to combine them to skeletons for individual person, top-down approaches convert the multi-person problem into a single-person problem by first detecting all persons in the image and then processing them independently. In bottom-up methods the information from the image is limited to the body parts that are visible, therefore often a post-processing step is used to complete the partial skeletons, in combination with wrong joint to person assignments this can lead to duplicate detections of persons. Furthermore the limiting factor for bottom-up approaches is the resolution of the body part maps, which usually is a compromise to allow enough distance between neighboring body parts and limiting the required memory and computation.

Over the course of the project we implemented multiple existing 2d and 3d single- and multi-person human pose estimation approaches in the same framework as baselines.

- **CoarseToFine [1]**
- **MargiPose [4]**
- **XNect [2]**
- **OpenPose [6]**

CoarseToFine is a bottom-up single person 3d human pose estimation method that extends the idea of predicting Gaussian heatmaps for each joint of the skeleton from a 2D image to a 3D volume by lifting an initial 2d heatmap to 3d using multiple hourglass modules. The joints can then be extracted from the heatmaps. While the representation as a 3d volumetric heatmap is appealing due to the ability to derive uncertainty from it, the required memory is very high if the working range is large.

MargiPose [4] tackles this limittion by replacing the volumetric representation for the joint probabilites with 3 marginal projections of the distribution to reduce the memory and computational requirements.

XNect follows a different approach, in a first step in addition to 2d joint heatmaps and part affinity fields (PAF), relative 3d offsets between all joints and their direct neighbors are estimated from the image. Using the PAFs a greedy assignment strategy based on OpenPose [6] is applied to link the joints to individual persons. Then the 2d poses of each person are lifted to 3d using a deep neural network, by also using the predicted 3d offsets between joints.

In addition we implemented two novel methods for 3d multi-person human pose estimation

We extend MargiPose, which was designed for the single-person usecase, for the case of multiple person per scene, by adding part affinity field from OpenPose to split the predicted joints into multiple persons and replacing the soft-argmin function to retrieve the peaks from the heatmap with different approaches (argmin, soft-argmin, meanshift, fitting a Gaussian mixture model to the heatmap). One limitation of this approach is that the projections of the joints from multiple people can potentially overlap in one or two of the projections and therefore cause interference and degradation of the accuracy.

5.2.2.2.2 Multi-person dataset from single-person dataset

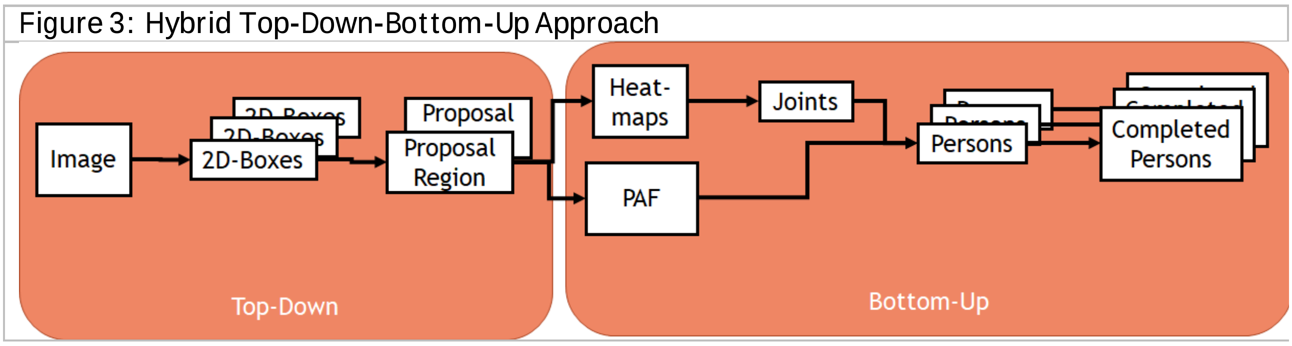
Most existing datasets contain only a single person moving in a small work space of 5x5m and are therefore not comparable with an application in the AD context.

With the aim of better generalization of the models we experimented with training the 2D and 3D parts of the networks on multiple datasets at the same time. As currently there exist much more 2D datasets than 3D annotated data, this mostly benefits the generalization of the 2D part. Nearly all datasets with available 3D annotations are captured indoors with a static background or greenscreen setup. We therefore added an data augmentation method similar to [3], which segments the captured persons from the background and allows the recombination with different backgrounds.

Using the data augmentation method it is possible to combine the individual recordings of multiple persons in a single image. This way it is possible to create a multi-person dataset from single person recordings. If no accurate depth data is available in the dataset the recombination in the cases of close proximity of body parts from different persons can lead to errors, we therefore investigated different methods to prevent these artifacts. To further augment the existing data that is mostly catured in small indoor space, we add background augmentation from greenscreen or static background.

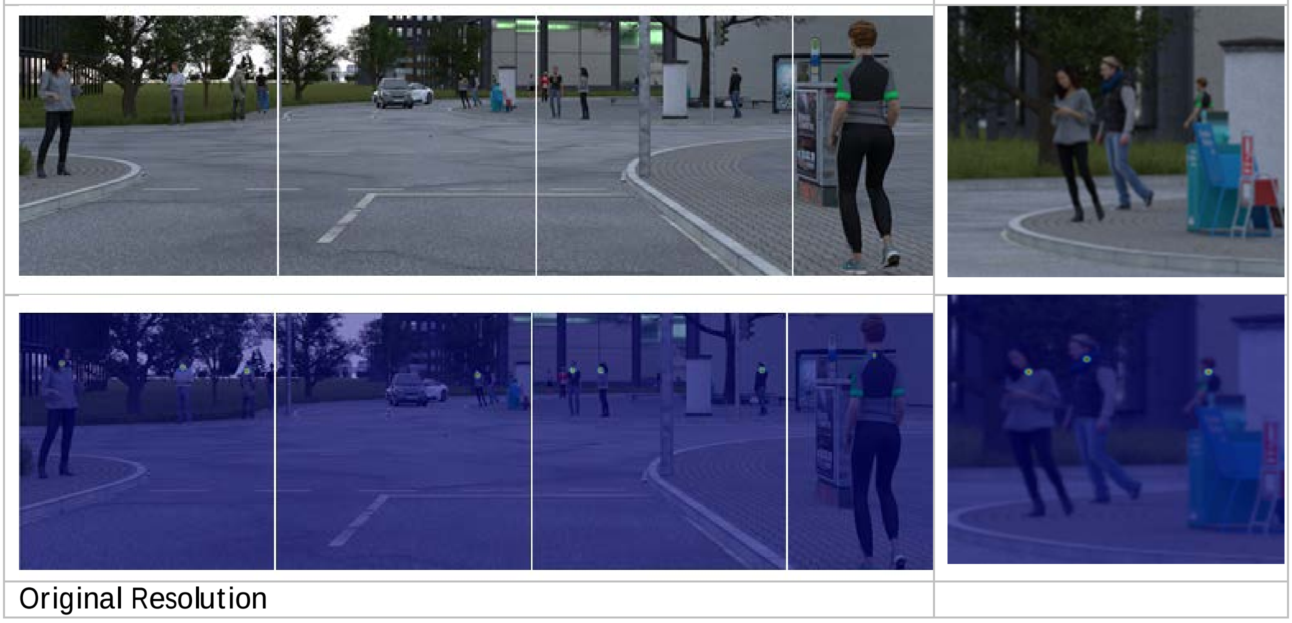
5.2.2.2.3 Hybrid Top-Down-Bottom-Up Approach

Figure 3: Hybrid Top-Down-Bottom-Up Approach



During our experiments we noticed two main drawbacks of the current approaches. The first one is related to the network architecture. Using the previously implemented blending method to generate a multi-person dataset from a single person dataset we created a synthetic dataset that is more similar to autonomous driving datasets with respect to the distribution of 3d position of the persons, but with a high variety of poses common in single person datasets. By evaluating the previously trained methods on this new dataset we found a significant performance gap, especially for persons further away from the camera. One reason for the poor performance is the domain gap between existing indoor datasets and outdoor applications like autonomous driving. By training our models on the KIA dataset the performance could be increased, but for further away poses we still observe a fast drop in the detection of joint positions and limb information (part affinity fields). The main reason for the limited performance on far away persons is the output resolution of the network branches used for joint and limb detection, as most existing networks downsample the input resolution of the image e.g. 512x512 by a factor of 8 to only 64x64 pixels. In contrast HRNet [5] proposes a different architecture to maintain a high resolution map throughout the network, while limiting the memory demand to enable efficient training on a single GPU. We therefore provide a new baseline following their architecture design and allow for different output resolutions up to input resolution. Due to memory constraint of todays GPUs it is not possible to increase the input size of these networks above 512x512 pixels. Additional experiments show that even when processing the input image with the original resolution in tiles of 512x512, several joints of persons standing far away from the camera are not detected, but when we run the network on tiles of a 4x super resolution image the joint detection performance keeps increasing.

Figure 4: Scale difference of pedestrians in the autonomous driving context.



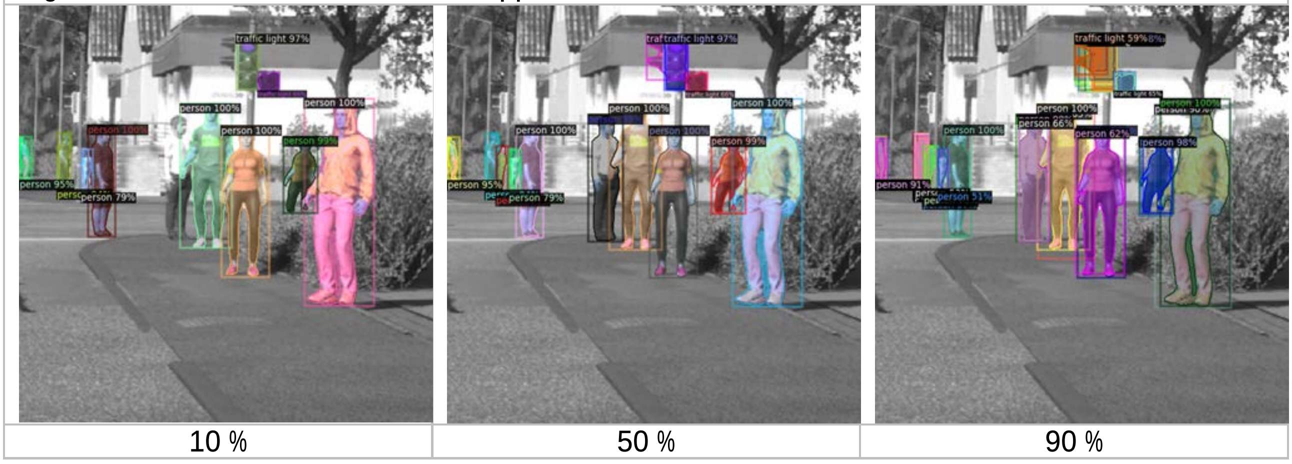
Original Resolution

This suggests that for bottom-up heatmap based joint and limb detection frameworks a super resolution can improve the results, but would increase the computational and memory requirements to a level which is not feasible anymore for real-time applications. We therefore started to work on a hybrid top-down bottom-up approach to first detect potential person/area-of-interest candidates in the original image and then zoom-in to process the candidates on an optimal resolution level. Initial experiments suggest a comparable performance to tile based processing, with a significantly reduced runtime. In contrast to end-to-end approaches commonly used in top-down approaches we believe that an bottom-up approach for each candidate can naturally produce multi-modal solutions and provide a better measure of uncertainty.

The second bottleneck is the matching of joints to persons, while the proposal based top-down preselection and instance segmentation help in reducing the combinatorial problem and the higher resolution also improves the part affinity fields, the greedy algorithm used in most approaches is in difficult cases often not able to find the correct solution. This problem remains even for proposal based approaches as the 2d projections of persons standing in groups often overlap significantly. Here we started investigating global optimization based approaches, which are more expensive, but can provide a guarantee to find the optimal solution. We aim to formulate a combined problem for overlapping candidates, find a globally consistent solution and include the matching uncertainty into the overall pose uncertainty.

For th top-down appraoches we noticed that depending on the hyperparamters used in the non-maximum suppression for the 2d bounding box detection some of the persons were not detected as the overlap with other higher scoring bounding boxes is too large, see Fig. 5. We therefore join the areas of the overlapping boundingboxes in the one with the highest resolution and combine the bottom-up predictions of joint positions and PAF, we then tested different methods to extract the most likely combination of limbs to form persons. In a next step we feed the often uncomplete 2d poses into a seperate network to predict the missing joints and lift the 2d pose to 3d similar to [2]. We noticed that this sometimes revealed persons that were previously suppressed in the 2d detection stage. We also noticed that in some cases the joints of a single person would not be connected correctly by the assignment process and therefore result in two partial skeletons for the same person, which are then completed by the next step and would naturally overlap, we are therefore investigating different strategies to detect these cases before or after completion and join them back together. We further looked into an alternative PAF representation as the original version only encoded the direction between joints two joints with the same orientation in a crowded area might be mismatched. We tried an alternative parameterization proposed in [7] where in addition to the direction the distance between the joints is encoded in the affinity field. While the different magnitude of the PAF vectors lead to a smoother field towards the head of the bone the encoded distance sometimes helps to disambiguate difficult cases.

Figure 5: Different non-maximum suppression thresholds and their effect



10 %

50 %

90 %

In the case of estimation the 3d human pose from a single RGB image the recovery of the absolute pose of the persons with respect to the camera is non-trivial. One way to resolve the ambiguity between the distance and size of a person is to assume a fixed size for all pedestrians. As this assumption clearly does not hold for all pedestrians e.g. children, the performance can vary strongly. In the use case of autonomous driving we usually have the front camera mounted on the car in a fixed position, assuming the car keeps level to the ground we can use this geometric knowledge to estimate the distance to each person by looking at the vertical position of their foot joints where they have contact with the groundplane. In practice the car is suspended over the ground with its four wheels and there are considerable pitch motions during the acceleration of the vehicle which can lead to large errors in depth estimation. Alternatively depth information can be learned from the context of the scene either using existing RGB 3d object detectors or monocular depth prediction approaches. Some approximate values are given in Tab. 1 for comparison. As the error for monocular depth is measured over the whole scene the values are not directly comparable to the object related appraoches, the real value should be more similar to them.

Table 1: Approximate absolute depth estimation errors	
Approach	Approximate Error [m]
Fixed human height (10 cm error)	0.7
Fixed human height (50 cm error)	3.5

Table 1: Approximate absolute depth estimation errors	
Known groundplane (5° error)	-
SotA 3D Object Detector	0.6
SotA Monocular Depth	1.6

5.2.2.3 Evaluation

We compare 3 different approaches on the valid/test tranches MV 4-7. We provide three different metrics, the mean per joint position error (MPJPE) measure the continous sum of errors between predicted and groundtruth relative poses. The percentage of correct keypoints (pCK@.5h) thresholds the MPJPE at 0.5 times the size of the distance between the neck and head joints. The mean distance (mD) measures the absolute distance between the predicted and groundtruth root joint (hip).

Method	MPJPE [m]	pCK@.5h [m]	mD [m]	Runtime [s]
Top-Down	N.A.	N.A.	N.A.	N.A.
Bottom-Up	N.A.	N.A.	N.A.	N.A.
Hybrid	N.A.	N.A.	N.A.	N.A.

5.2.3 Conclusion

3D human pose estimation for the task of autonomous driving introduces special challenges that have previously not been a focus in the research. The layout of the scene captures by the front facing camera leads to a wider range of sizes of persons and in urbsan environments pedestrians tend to form crowds which pose a special challenge to existing bottom-up and top-down appraoches. We therefore developed a hybrid method that combines the advantages of both approaches and runs with a afforable computational budget by being able to capture the regions of interest in the image efficiently. Experiments show....

5.2.3.1 References

[1] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G. Derpanis, Kostas Daniilidis, **Coarse-to-Fine Volumetric Prediction for Single-Image 3D Human Pose**, CVPR 2017

[2] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Mohamed Elgharib, Pascal Fua, Hans-Peter Seidel, Helge Rhodin, Gerard Pons-Moll, Christian Theobalt, **XNect: Real-time Multi-person 3D Human Pose Estimation with a Single RGB Camera**, CVPR 2019

[3] Mehta, D., Sotnychenko, O., Mueller, F., Xu, W., Sridhar, S., Pons-Moll, G., & Theobalt, C.. **Single-shot multi-person 3d pose estimation from monocular rgb**, 3DV 2018.

[4] Nibali, A., He, Z., Morgan, S., & Prendergast, L., **3d human pose estimation with 2d marginal heatmaps**, WACV 2019

[5] Ke Sun*, Bin Xiao*, Dong Liu, Jingdong Wang, **Deep High-Resolution Representation Learning for Human Pose Estimation**, CVPR 2019

[6] Z. Cao, G. Hidalgo, T. Simon, S. -E. Wei and Y. Sheikh, **OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields**, TPAMI 2019

[7] Kreiss, S., Bertoni, L., & Alahi, A., **Pifpaf: Composite fields for human pose estimation,** CVPR 2019

[8] Li, J., Wang, C., Zhu, H., Mao, Y., Fang, H. S., & Lu, C., **Crowdpose: Efficient crowded scenes pose estimation and a new benchmark**, CVPR 2019

[9] Golda, T., Kalb, T., Schumann, A., & Beyerer, J., **Human pose estimation for real-world crowded scenarios**, AVSS 2019

5.3 E1.5.3 Final: Algorithmus zur 3D-Posenschätzung aus RGB-und Tiefe-Daten (zur Veröffentlichung)

5.3.1 Formal Classification

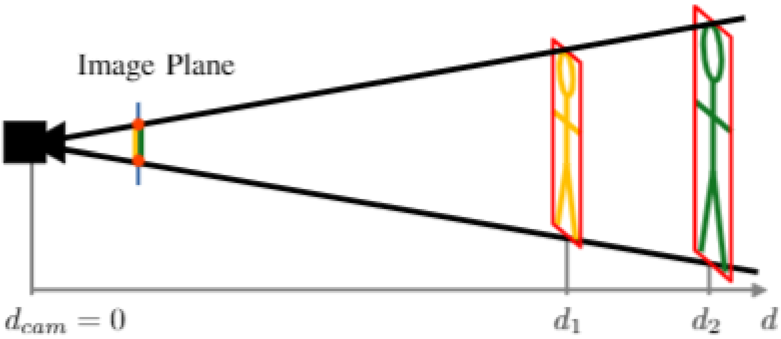
Criteria	Classification according to VHB
Type of result	*Code*
Group/Cluster	
Type of content	*DL Model*
Classification level	*PU*

5.3.2 Description of the result

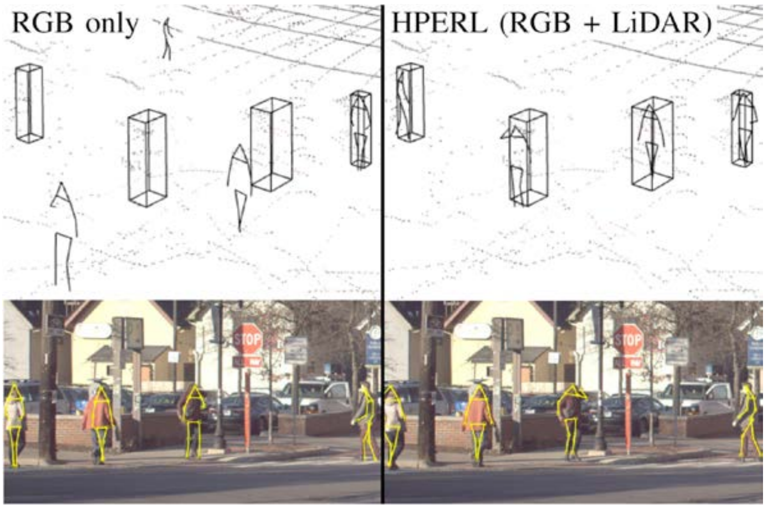
5.3.2.1 Introduction

Human pose estimation can help with understanding pedestrian's behaviour, identify corner cases and do data coverage analysis in the dataset. In the current state-of-the-art predicting the position of the joints in 2D - called 2D human pose estimation - is very common. Less common is 3D human pose estimation, where the joint positions are predicted in 3D space.

Like for object detection there are multiple sensor setups that can be used. It is possible to use only RGB cameras or to fuse RGB and LiDAR data. While RGB only approaches have been well explored, there is a lack of fusion approaches for human pose estimation in the current literature for the state-of-the-art. However, LiDAR has a strong advantage over RGB only since it does not suffer from depth ambiguity. Depth ambiguity describes the problem that two objects with different sizes, which have also different distances to the camera appear to be similar-sized, projected to the camera plane. Thus without knowing the exact size of an object or the distance it is impossible to correctly infer the other.

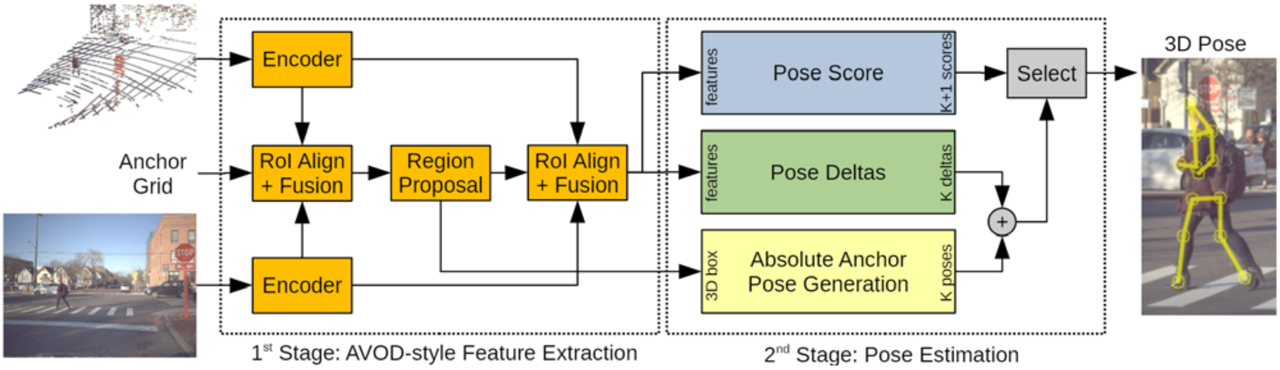


Thus, we propose a methodology to predict the 3D Human Pose from RGB and LiDAR data. We use the fusion pattern of the 3D detector AVOD and integrate it into the LCR-Net human pose estimator. By doing this we could achieve great results. The approach was published as "HPERL: 3D Human Pose Estimation from RGB and LiDAR" in ICPR 2020.



5.3.2.2 Approaches

We identified that LCR-Net and AVOD both have a Faster-RCNN like architecture and thus are well suited for combining. In our HPERL approach we use the first stage of AVOD for feature extraction and the second stage of LCR-Net for 3D Pose estimation. The resulting architecture is visualized in the figure.



Due to a lack of 3d joint position annotations, we could not directly train the model. Instead we derived a new loss function, that trains the model on two losses for the 3d pose. One component of the loss optimizes the position of the skeleton to fit within the ground truth 3d bounding box and the other loss minimize the loss of the joints projected to 2D to the ground truth 2D pose annotations available on the PedX Dataset.

5.3.2.3 Experiments

Unfortunately the data produced in KI-Absicherung is unsuitable for our approach as it needs RGB, LIDAR and human pose annotations. All data separately exists within the project, but only BIT provides LiDAR, while only Mackevision provides human pose annotations. Thus we had to use public datasets for our experiments in order to validate our approach.

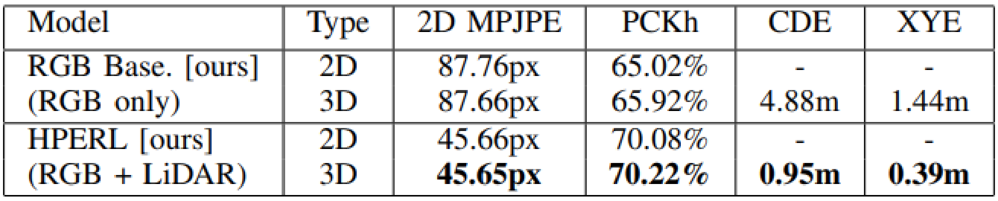
The dataset used is the PedX Dataset with Camera, LiDAR, 3D bounding box annotations and 2D pose annotations. Due to this limitation we used the implicit loss for training the 3D human pose as described in our approach.

We validated our baseline RGB only detector to perform similar to the state-of-the-art model LCR-Net on which our approach is based (Table 2 and Table 3). Then we were able to compare our RGB baseline approach against Fusion of RGB and LiDAR. The accuracy of the joint position in 2D improved slightly (2D MPJPE and PCKh), while the 3d accuracy of the center distance error (CDE) and the position error on the ground plane (XYE) significantly improved from 4.88m to 0.95m and from 1.44m to 0.39m respectivley (Table 1).

In the ablation studies (Table 4) we show that fusion, fusion method, pretraining and augmentation have an impact on the performance of the model. The biggest lift in performance comes from using fusion vs no fusion and then the correct fusion and pooling operation are the second largest contributor.

TABLE I

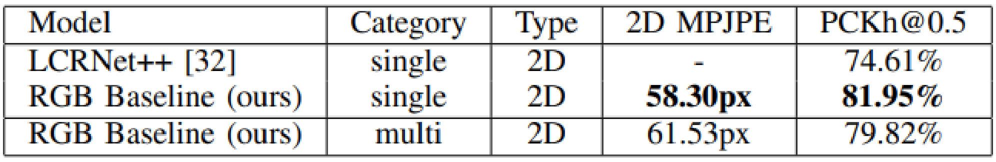
Comparison of RGB Baseline vs HPERL on PedX. LiDAR significantly improves the precision of 3D location (1/5 CDE, 1/3 XYE). 2D results improve slightly (MPJPE and PCKh@0.5).



Model	Type	2D MPJPE	PCKh	CDE	XYE
RGB Base. [ours]	2D	87.76px	65.02%	-	-
(RGB only)	3D	87.66px	65.92%	4.88m	1.44m
HPERL [ours]	2D	45.66px	70.08%	-	-
(RGB + LiDAR)	3D	**45.65px**	**70.22%**	**0.95m**	**0.39m**

TABLE II

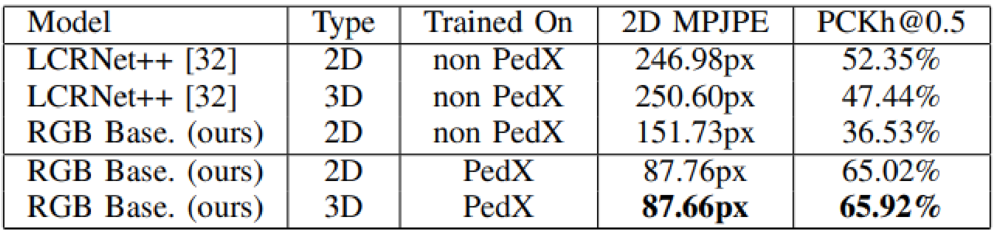
RGB Baseline (inspired by LCRNet++) Verification on MPII



Model	Category	Type	2D MPJPE	PCKh@0.5
LCRNet++ [32]	single	2D	-	74.61%
RGB Baseline (ours)	single	2D	**58.30px**	**81.95%**
RGB Baseline (ours)	multi	2D	61.53px	79.82%

TABLE III

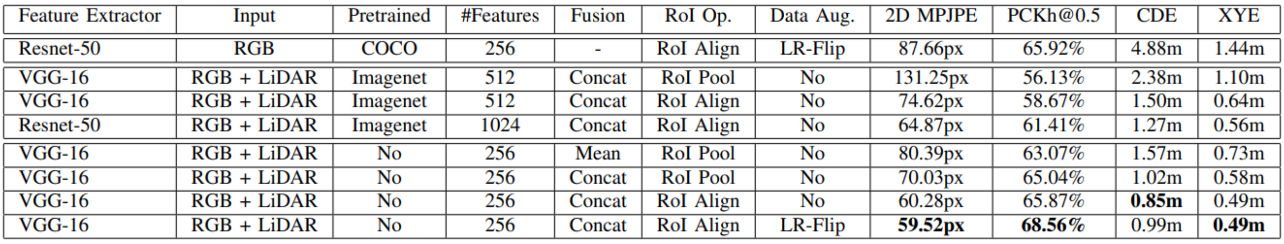
RGB Baseline (inspired by LCRNet++) Verification on PedX.



Model	Type	Trained On	2D MPJPE	PCKh@0.5
LCRNet++ [32]	2D	non PedX	246.98px	52.35%
LCRNet++ [32]	3D	non PedX	250.60px	47.44%
RGB Base. (ours)	2D	non PedX	151.73px	36.53%
RGB Base. (ours)	2D	PedX	87.76px	65.02%
RGB Base. (ours)	3D	PedX	**87.66px**	**65.92%**

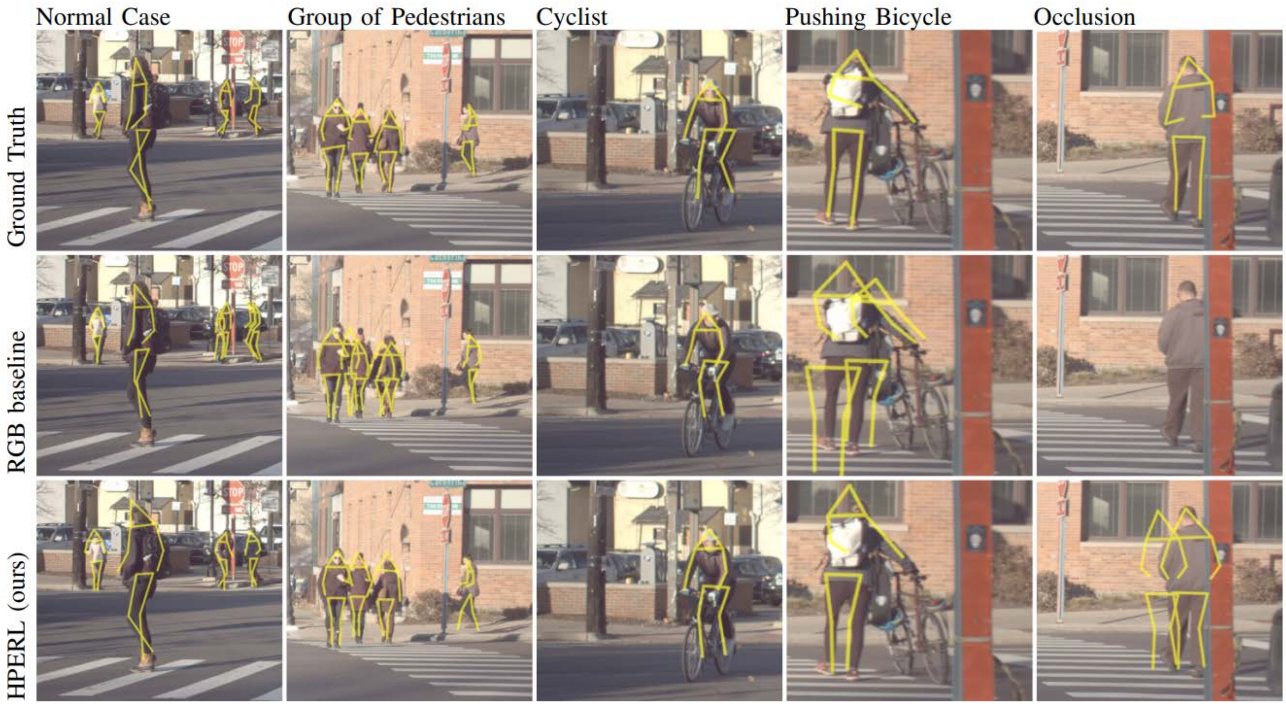
TABLE IV

Ablation studies of HPERL on the PedX dataset. Feature extractor, pre-training, fusion, RoI operation and data augmentation were varied to determine the impacts on the 3D pose estimation on PedX. The biggest impact is due to adding LiDAR.



Feature Extractor	Input	Pretrained	#Features	Fusion	RoI Op.	Data Aug.	2D MPJPE	PCKh@0.5	CDE	XYE
Resnet-50	RGB	COCO	256	-	RoI Align	LR-Flip	87.66px	65.92%	4.88m	1.44m
VGG-16	RGB + LiDAR	Imagenet	512	Concat	RoI Pool	No	131.25px	56.13%	2.38m	1.10m
VGG-16	RGB + LiDAR	Imagenet	512	Concat	RoI Align	No	74.62px	58.67%	1.50m	0.64m
Resnet-50	RGB + LiDAR	Imagenet	1024	Concat	RoI Align	No	64.87px	61.41%	1.27m	0.56m
VGG-16	RGB + LiDAR	No	256	Mean	RoI Pool	No	80.39px	63.07%	1.57m	0.73m
VGG-16	RGB + LiDAR	No	256	Concat	RoI Pool	No	70.03px	65.04%	1.02m	0.58m
VGG-16	RGB + LiDAR	No	256	Concat	RoI Align	No	60.28px	65.87%	**0.85m**	0.49m
VGG-16	RGB + LiDAR	No	256	Concat	RoI Align	LR-Flip	**59.52px**	**68.56%**	0.99m	**0.49m**

In a qualitative comparison below of performance between the RGB baseline and HPERL the poses are depicted in yellow. In common scenarios shown on the left, both algorithms detect the pedestrians, but the baseline struggles with false positives at multiple depths. Even rare cases as humans riding bycicles are well detected by both methods. Pushing a bicycle however causes false positives for RGB baseline and an imprecise detection for HPERL. Partial occlusions are difficult for both approaches. However HPERL is able to detect the pedestrian but at the cost of a false positive.



5.3.3 Conclusions

In the presented work, we discovered, that fusion of LiDAR and RGB can indeed improve the performance of 3D human pose estimation in the automotive context. Our approach has a much better depth precision than the RGB only network. With respect to the safety of the pedestrians this makes our approach better than prior state of the art, because a 4.88 meter error in depth perception is in acceptable. The 0.95 meters achieved by our approach is a significant improvement.

One downside is that due to the lack of datasets with this specific sensor setup and the pose annotations research and evaluation is difficult and limits the application. This means, we were also not able to test the approach on the KI-Absicherungs data, as it is incompatible. We hope that our findings motivate the development and publication of datasets with the suited sensor and annotation combination, enabling more research in this promising direction.

The full work has been published in: Fuerst et al. "HPERL: 3D Human Pose Estimation from RGB and LiDAR" in ICPR 2020. (see https://arxiv.org/pdf/2010.08221.pdf)

5.4 E1.5.4 Final: nur projektintern für KI Absicherung verfügbar

5.5 E1.5.5 Final: nur projektintern für KI Absicherung verfügbar

5.6 E1.5.7 Final: nur projektintern für KI Absicherung verfügbar