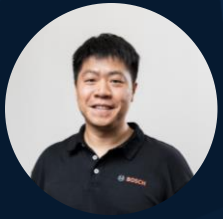
BOSCH

AI数据驱动工程 - 如何在大模型时代保障高阶辅助驾驶感知系统的安全性

季康, 博世车辆运动智控事业部



2024 MathWorks
中国汽车年会

Agenda

Motivation & Introduction

Data-Driven Testing

Data in Machine Learning

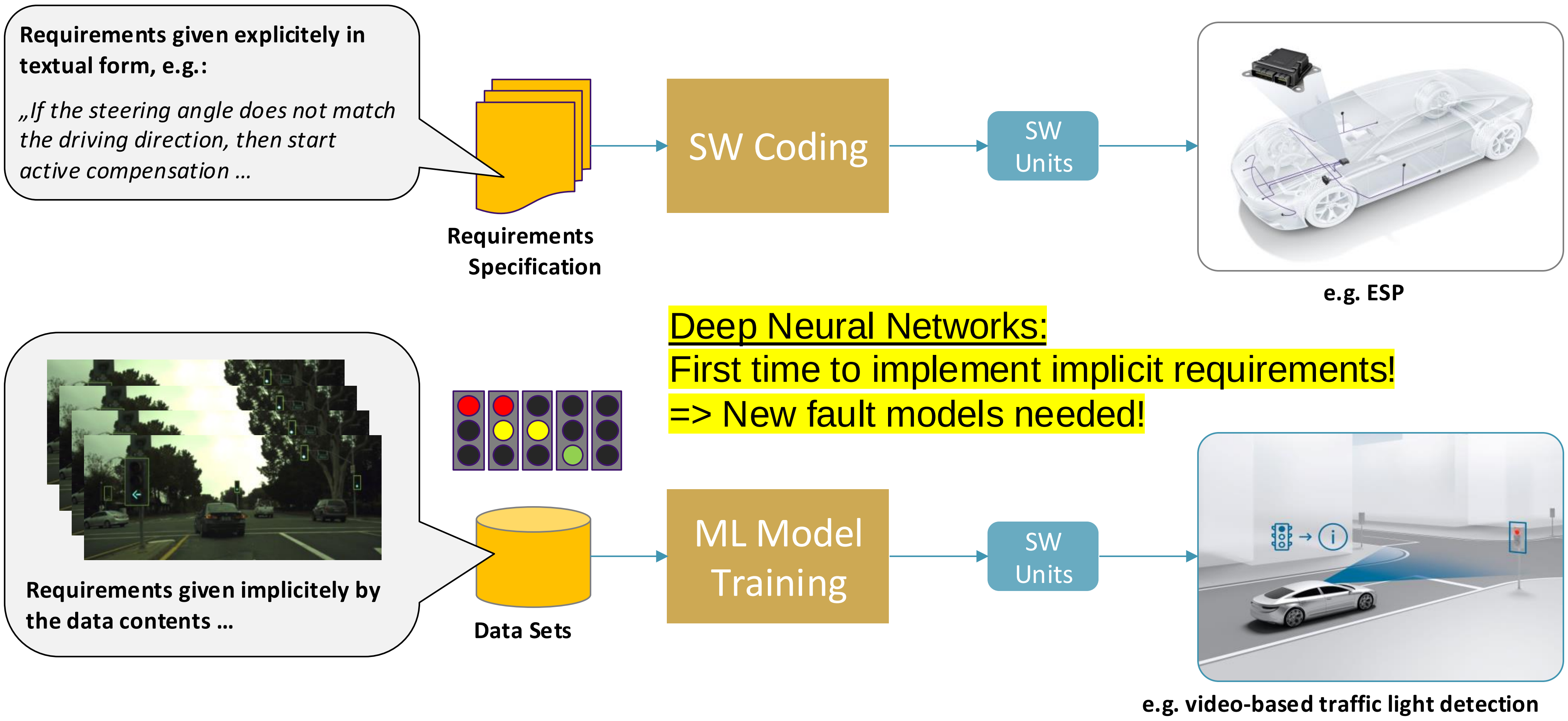
Data-Driven Engineering (DDE) as Solution

Automation of DDE Process Workflows

Demo on DDE Process Workflows & Outlook

Credits to: Andreas Albrecht (XC-AS/EDL2) & Zhuang Lin (XC-AS/EDL1)

Classic SW Development versus Machine Learning



Requirements given explicitely in textual form, e.g.:

„If the steering angle does not match the driving direction, then start active compensation …

Requirements Specification
SW Coding
SW Units
e.g. ESP

Deep Neural Networks:

First time to implement implicit requirements!

=> New fault models needed!

Requirements given implicitely by the data contents …

Data Sets
ML Model Training
SW Units
e.g. video-based traffic light detection

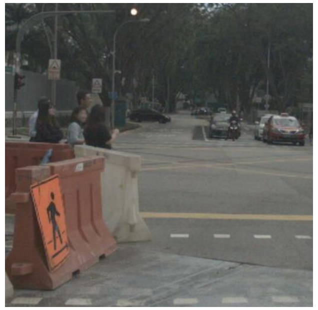
ML models learn their I/O-behavior from the training data. We need to design our data sets to implicitely define the ML function!

AD: Automated Driving DDE: Data-Driven Engineering DNN: Deep Neural Network FoV: Field of View GT: Ground Truth ML: Machine Learning (connectionistic ML – esp. DNNs) ODD: Operational Design Domain TL: Traffic Light

Challenging Scenarios for Perception Systems

Construction sites with workers

Occluded pedestrians



Pedestrians with low contrast (at night)

Metal structures + rainy weather + bad lighting conditions

Complex urban scenarios

Rare unusual human poses

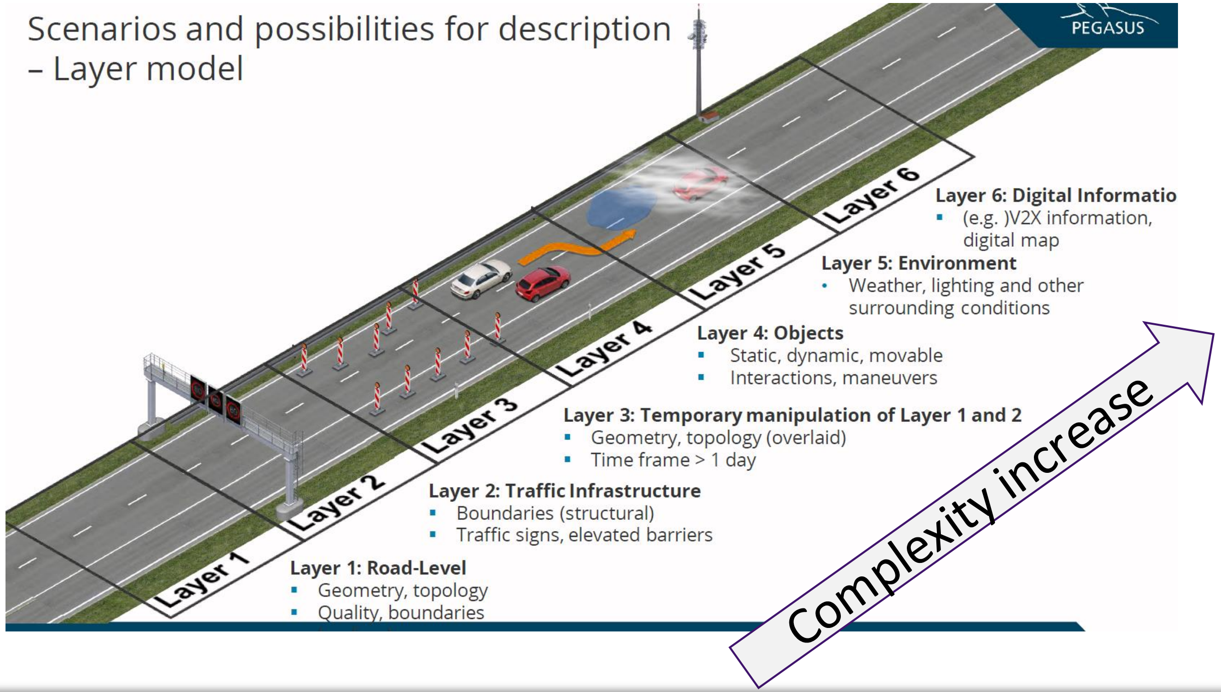
Dark tunnels + cyclists or pedestrians

Steel bridges, guard rails

Specific scenario constellations and environment conditions may trigger functional insufficiencies of our system under test.

ODD Model & Ontology are the Basis for Data Space Description

Defined Layer of influence parameter (Environment, Infrastructure, other road users)



Scenarios and possibilities for description – Layer model

Layer 6: Digital Informatio
- (e.g.)V2X information, digital map

Layer 5: Environment
- Weather, lighting and other surrounding conditions

Layer 4: Objects
- Static, dynamic, movable
- Interactions, maneuvers

Layer 3: Temporary manipulation of Layer 1 and 2
- Geometry, topology (overlaid)
- Time frame > 1 day

Layer 2: Traffic Infrastructure
- Boundaries (structural)
- Traffic signs, elevated barriers

Layer 1: Road-Level
- Geometry, topology
- Quality, boundaries

Complexity increase

Concept phase

Human experts shall be able to formulate scenarios in the fields terminology in **natural language.**

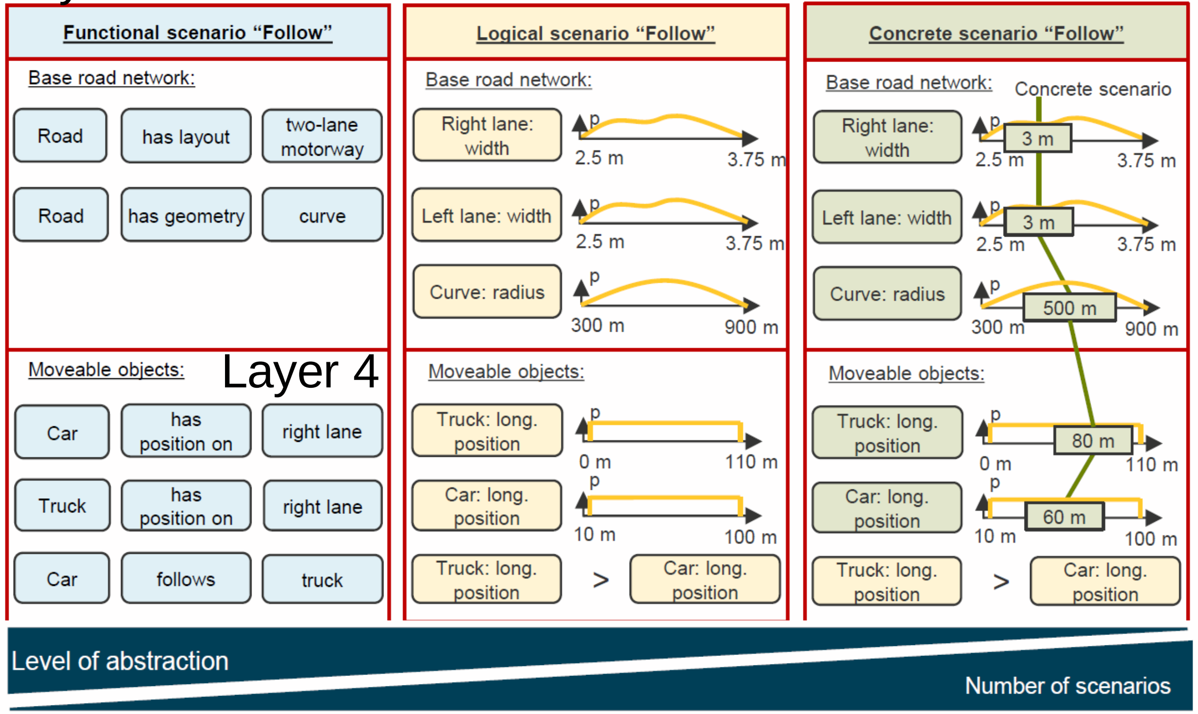
System development phase

Scenarios shall include the **parameter ranges** of the **state values** used for scenario representation.

Test phase

Scenarios shall be modeled via a **single representative** for each state value to ensure reproducibility.

Layer 1



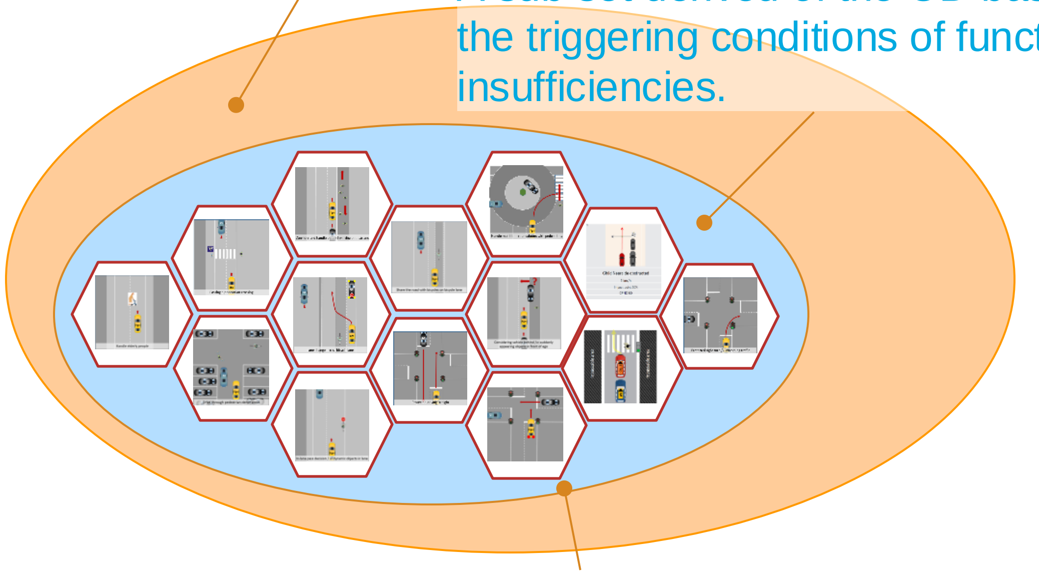
Layer 4

Ideally we will derive our domain-specific languages to describe our data space from a common ODD model & ontology

Gain Deep Understanding of the Operational (Design) Domain

Operational Domain (OD) for which an automated driving function is designed.

Operational Design Domain (ODD): A sub-set derived of the OD based on the triggering conditions of functional insufficiencies.



The border between OD and ODD has to be defined by SOTIF: What is in & what is out of ODD bounds.

ODD description by decomposition into ODD categories & items.

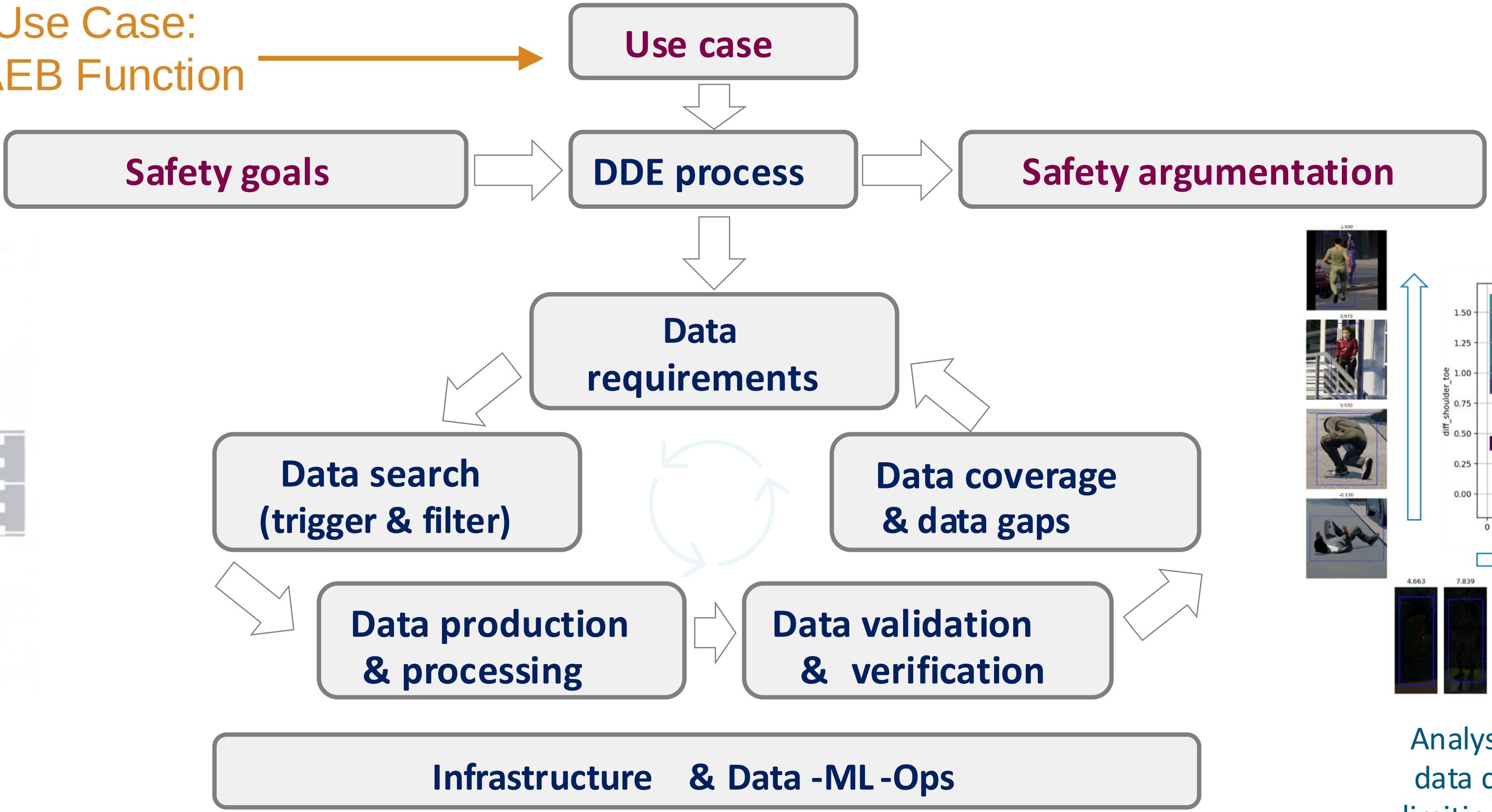
Scenario categories by SOTIF

	Safe / Non-hazardous	Unsafe / Hazardous
Known	Maximize number of known scenarios that the system can handle safely and test systematically	Eliminate, minimize or mitigate any potential risks and test systematically
Unknown	No further measures required (but consider them in testing as well)	Detect through analyses and systematic testing & data collection or generation

The ODD is decomposed into scenario classes & categories, which are analysed if they can trigger functional insufficiencies.

Systematically Derive Safety Arguments based on Data Coverage

First DDE PoC Use Case:
Vision-baed L2+ AEB Function


Use case

Safety goals

DDE process

Safety argumentation

Data requirements

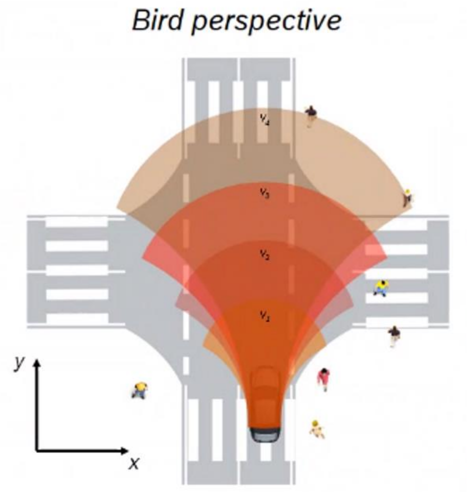
Data search (trigger & filter)

Data coverage & data gaps

Data production & processing

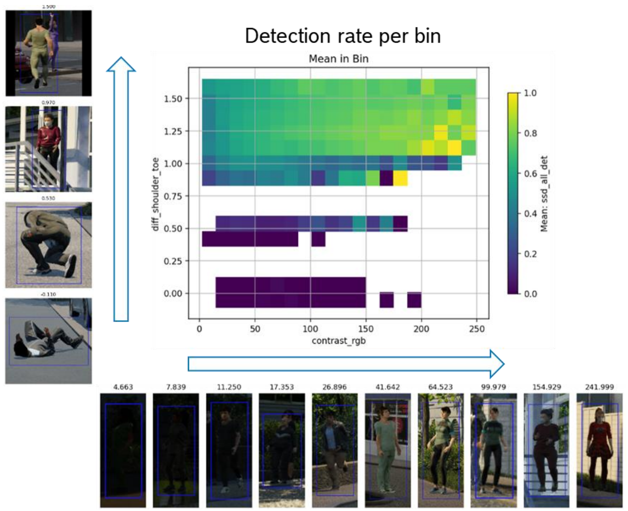
Data validation & verification

Infrastructure & Data -ML -Ops



Derive safety-related data requirements from system use case and Operational Design Domain (ODD)

Filter data pools for suitable samples in order to systematically increase test data coverage w.r.t. performance limiting factors to support a safety argumentation.



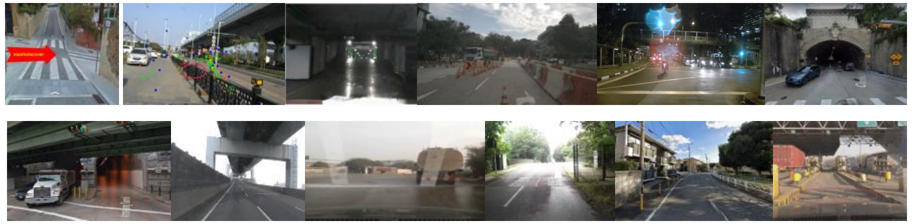
Analyse DNN performance versus data coverage w.r.t. performance limiting factors to find weaks spots

The DDE process supports systematic derivation & iterative refinement of data requirements to support a safety argumentation.

Verified Datasets with defined Coverage for Data-driven Testing

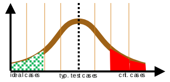
Selective search for desired scenarios or data features to compile representative test datasets with defined data coverage

Design datasets with defined coverage

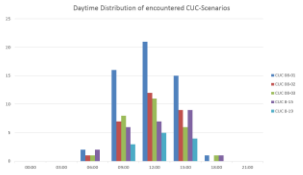


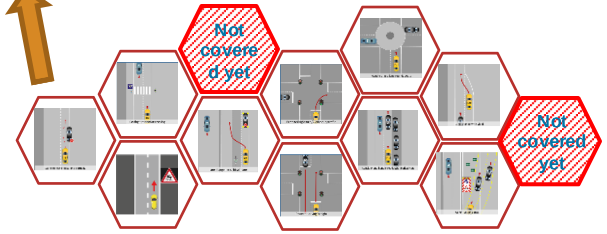
Filter for data items with desired features

Data pool with raw or partially labeled data



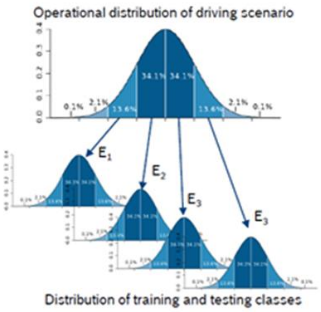
Raw sensor & meta data





Verified test dataset for a specific test goal

Data coverage statistics



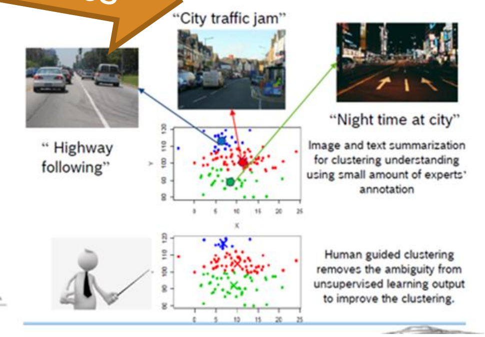


Image and text summarization for clustering understanding using small amount of experts' annotation

Human guided clustering removes the ambiguity from unsupervised learning output to improve the clustering.

Test Plan

Test Data

Test Data Link: … & **Contents:** …

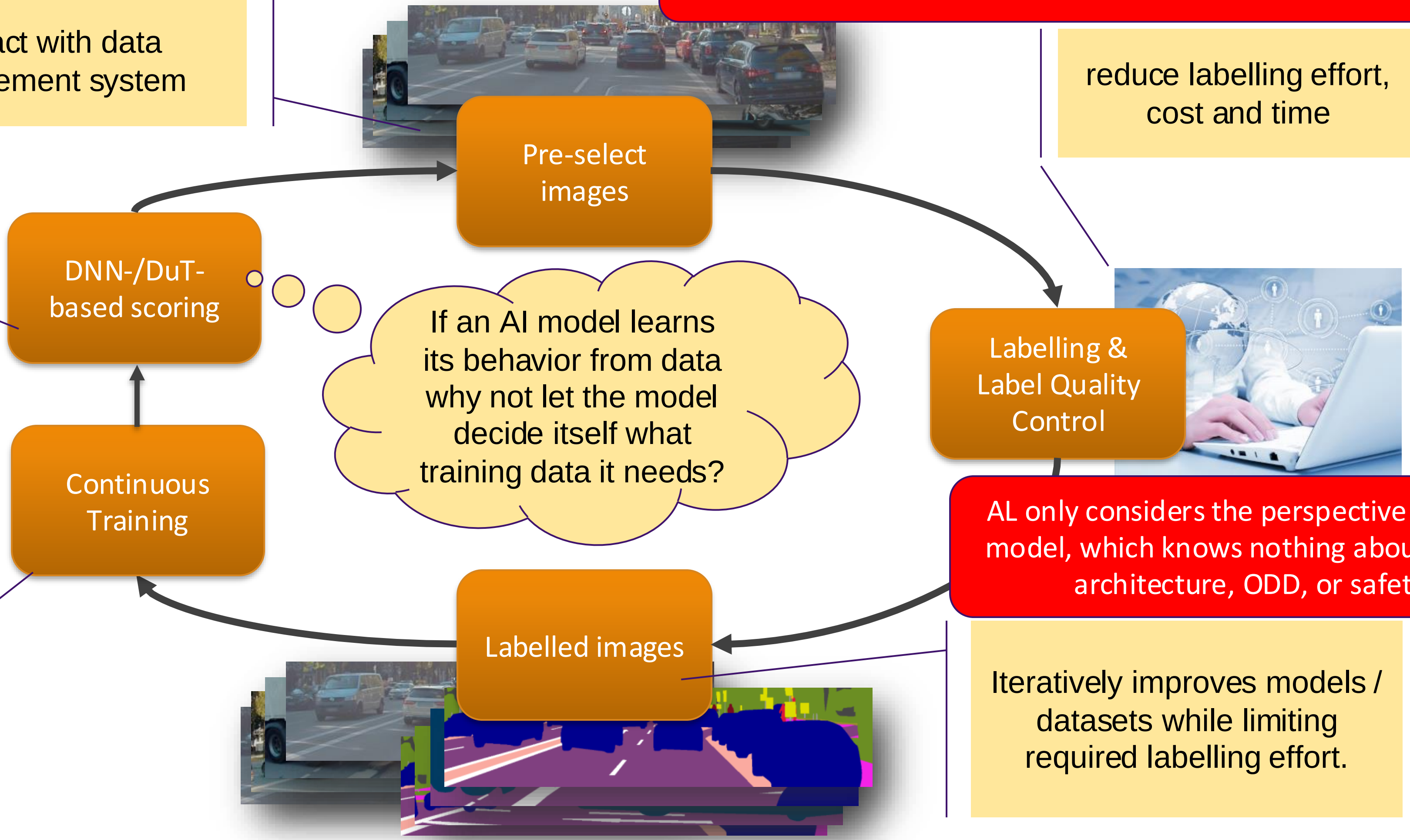
Test Suite 1

Test Suite 2

Execute test suites on the given test dataset

We want to have verified test datasets with defined coverage that are representative for the Operational Design Domain (ODD).

Verified Datasets with defined Coverage


Pre-select images
Labelling & Label Quality Control
Labelled images
Continuous Training
DNN-/DuT-based scoring
If an AI model learns its behavior from data why not let the model decide itself what training data it needs?

To provide sufficient ODD coverage we must consider a top-down system & safety engineering perspective.

Interact with data management system

reduce labelling effort, cost and time

Scan & score existing & incoming data

Scoring is biased focusing on the current decision boundaries of the DuT.

AL only considers the perspective of the AI model, which knows nothing about system architecture, ODD, or safety.

AL does not apply for testing.

Integrates closely with Continuous Training

Iteratively improves models / datasets while limiting required labelling effort.

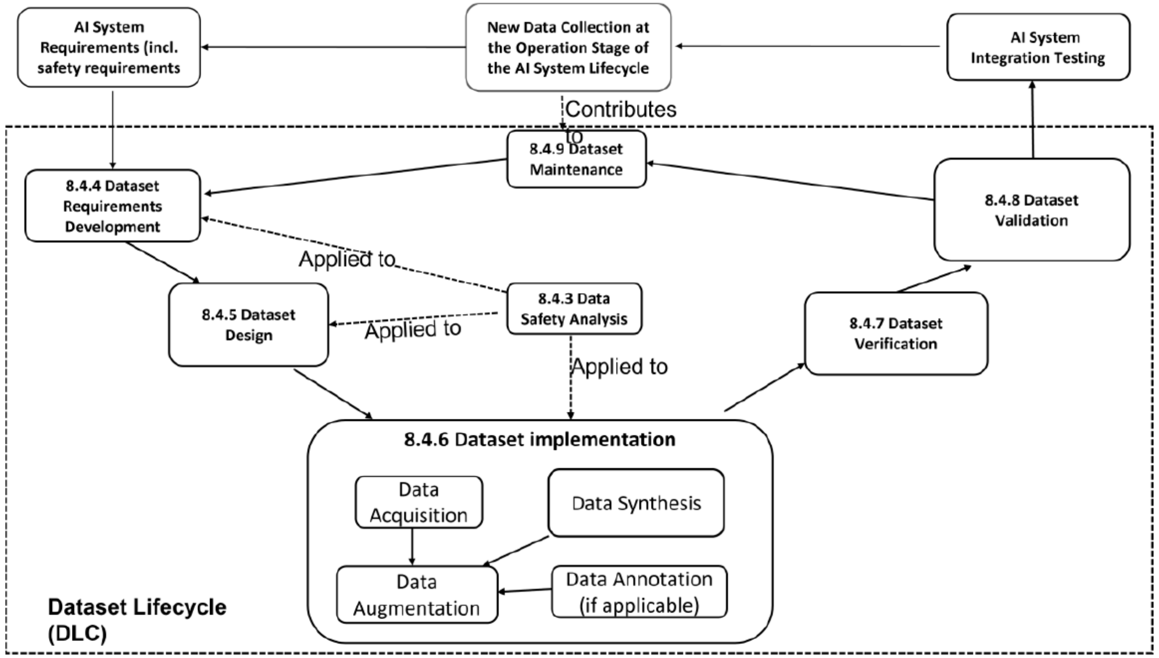
AL focuses only on the AI model's perspective, but does not consider the ODD or any system & safety engineering aspects.

AL: Active Learning AI: Artificial Intelligence DNN: Deep Neural Network DuT: Detector under Test

Iterative, Systematic Data-Driven Engineering Process Needed

AI Life Cycle Model with Data Life Cycle Model required

ISO PAS 8800 2022-09-20



AI System Requirements (incl. safety requirements

New Data Collection at the Operation Stage of the AI System Lifecycle

AI System Integration Testing

Contributes to

8.4.9 Dataset Maintenance

8.4.4 Dataset Requirements Development

8.4.8 Dataset Validation

Applied to

8.4.3 Data Safety Analysis

8.4.5 Dataset Design

Applied to

Applied to

8.4.7 Dataset Verification

8.4.6 Dataset implementation

Data Acquisition

Data Synthesis

Data Augmentation

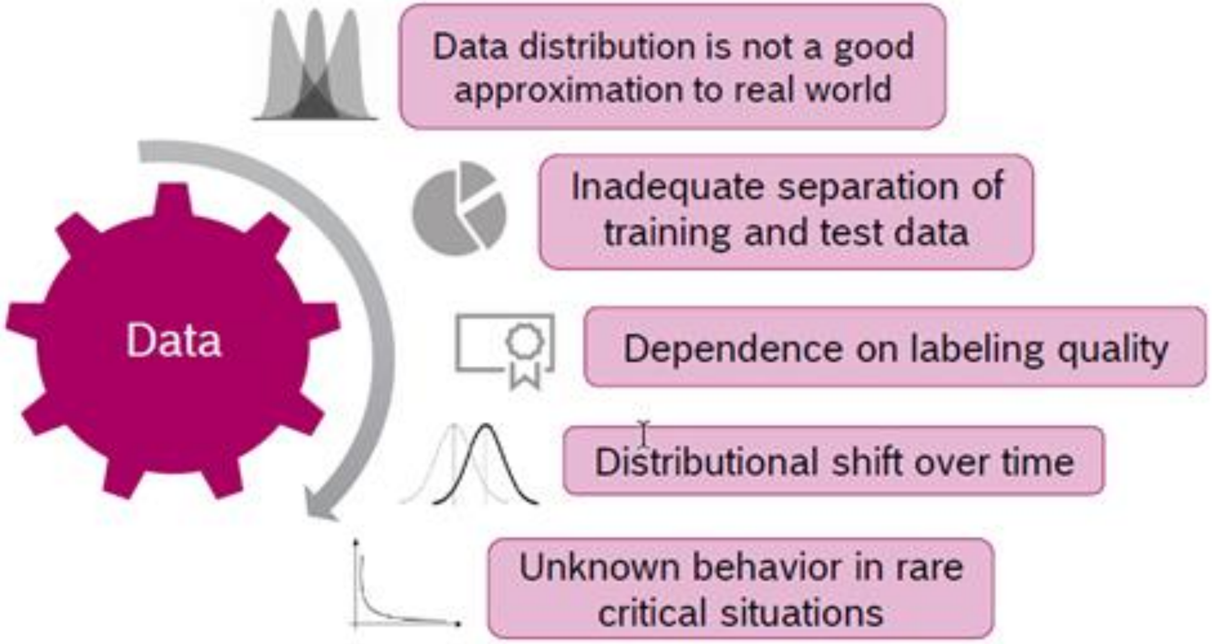
Data Annotation (if applicable)

Dataset Lifecycle (DLC)

Figure 10–2 — Dataset Lifecycle Model

Source: ISO/AWI PAS 8800:2023(E), Road Vehicles — Safety and artificial intelligence

AI safety conerns raised on data must be addressed



Data

Data distribution is not a good approximation to real world

Inadequate separation of training and test data

Dependence on labeling quality

Distributional shift over time

Unknown behavior in rare critical situations

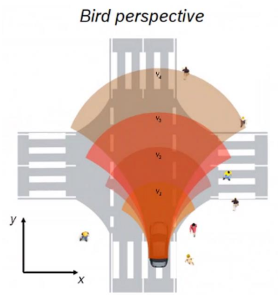
Source: KI Absicherung

Future safety argumentation must address safety concerns amongst others on data aspects raised for the respective AI system.

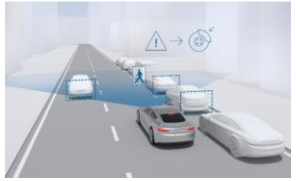
AI: Artificial Intelligence

Data-Driven Engineering brings in Top-Down SYS, ODD & Safety

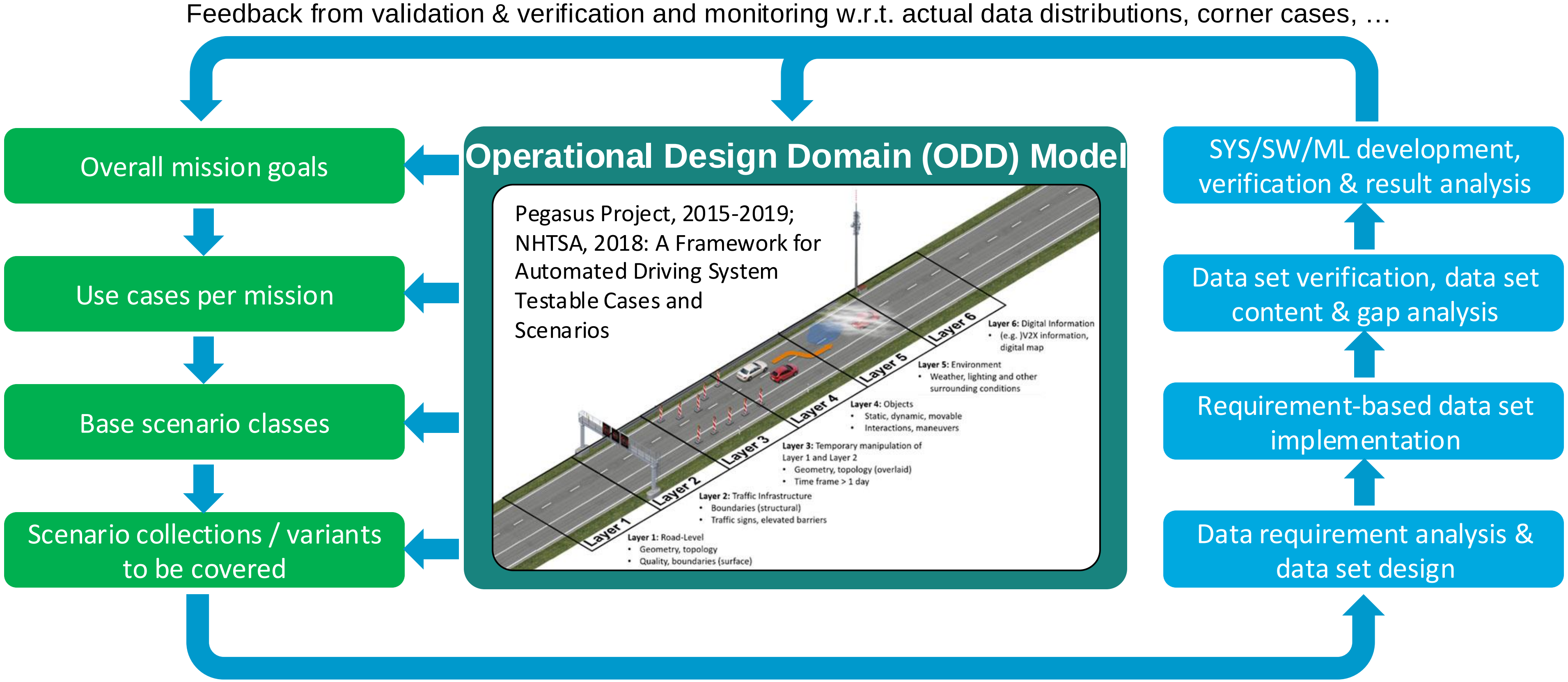
Drive from A to B



Detect collision-relevant objects



Emergency braking scenario: Occluded pedestrian steps Into driving corridor


Layer 1
Layer 2
Layer 3
Layer 4
Layer 5
Layer 6
Layer 1: Road-Level
• Geometry, topology
• Quality, boundaries (surface)
Layer 2: Traffic Infrastructure
• Boundaries (structural)
• Traffic signs, elevated barriers
Layer 3: Temporary manipulation of Layer 1 and Layer 2
• Geometry, topology (overlaid)
• Time frame > 1 day
Layer 4: Objects
• Static, dynamic, movable
• Interactions, maneuvers
Layer 5: Environment
• Weather, lighting and other surrounding conditions
Layer 6: Digital Information
• (e.g.)V2X information, digital map

Feedback from validation & verification and monitoring w.r.t. actual data distributions, corner cases, …

Overall mission goals

Use cases per mission

Base scenario classes

Scenario collections / variants to be covered

Operational Design Domain (ODD) Model

Pegasus Project, 2015-2019; NHTSA, 2018: A Framework for Automated Driving System Testable Cases and Scenarios

SYS/SW/ML development, verification & result analysis

Data set verification, data set content & gap analysis

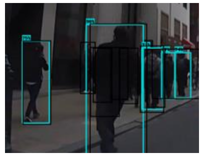
Requirement-based data set implementation

Data requirement analysis & data set design

Input: Desired scenarios, ODD elements with attributes incl. parameter ranges as search criteria

Unusual poses

Occlusion

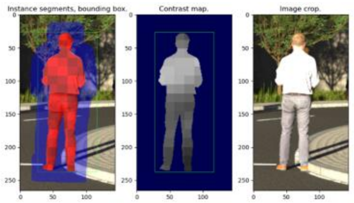


Low contrast or brightness

An aligned ODD model & ontology is the basis for deriving data requirements & criteria for data coverage on all design levels.

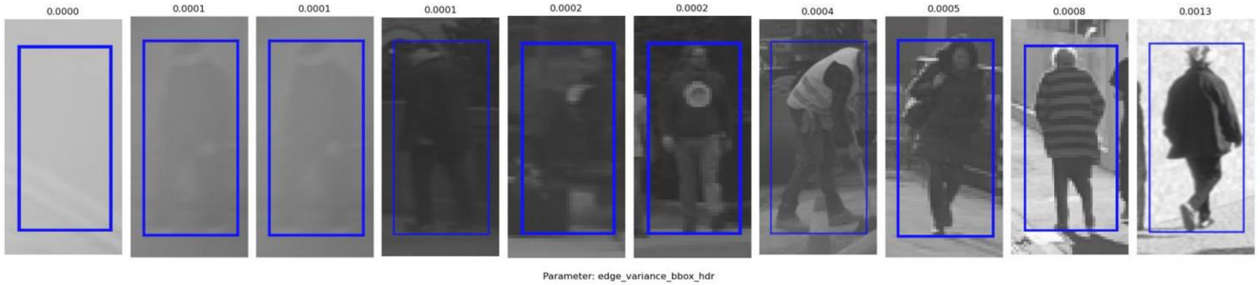
AD: Automated Driving DDE: Data-Driven Engineering DNN: Deep Neural Network FoV: Field of View GT: Ground Truth ML: Machine Learning (connectionistic ML – esp. DNNs) ODD: Operational Design Domain TL: Traffic Light

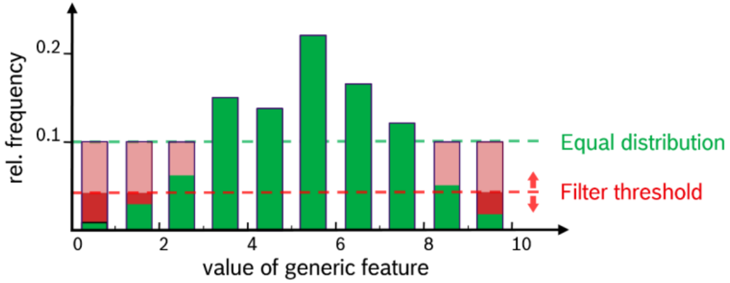
How to Specify Data Requirements in DDE



Example: RMS contrast along object contour as potential data feature candidate to be specified by a data requirement (focus on data contents)

Specify the relevant data feature space and cut it into discrete bins





Define a desired statistical distribution per data feature class or combinatorial class (e.g. min / max) to make it testable & find existing gaps

DDE Data Requirement

- Scenario class ID / data feature class ID

- **Qualitative** (data content-based) specification of ODD coverage => which scenario / data feature classes => derive filter criteria
- **What kind of scene/scenario** or data sample are we looking for?

- **Quantitative** specification of ODD coverage
- **How many frames or samples** of each scenario class resp. data feature class do we need?

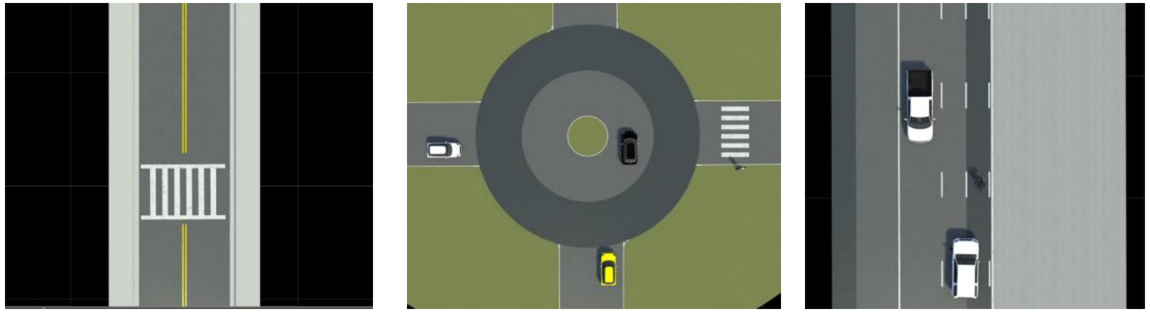
Data Set Design = set of data filter configurations (or configs for data augmentation / synthetization & simulation)

Verify data set / coverage with DDE Data Test Cases

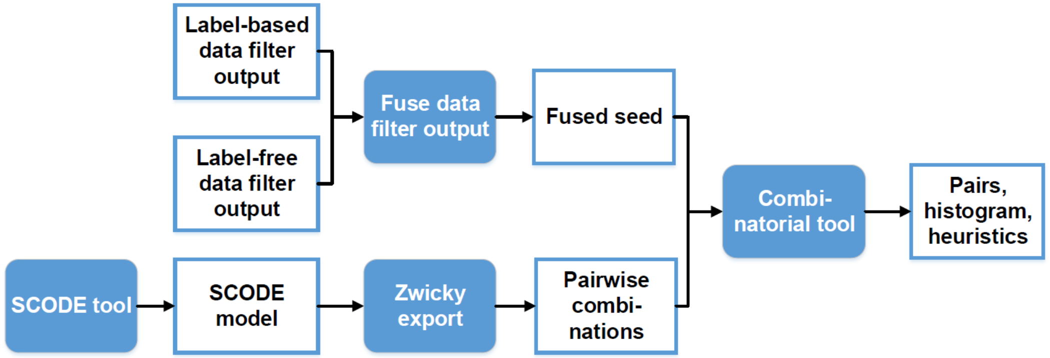
Requirement-based data set implementation & verification is a key step that can be automated using configurable data filters.

DDE: Data-Driven Engineering ID: identifier ODD: Operational Design Domain RMS: Root mean square
Software Val: (Cross-)Validation

Define Scenario Variations using Combinatorial Methods



Each scenario in the catalogue may occur in different variants – safety-relevant aspects can be derived analytically by combinatorial methods.



Label-based data filter output

Label-free data filter output

Fuse data filter output

Fused seed

SCODE tool

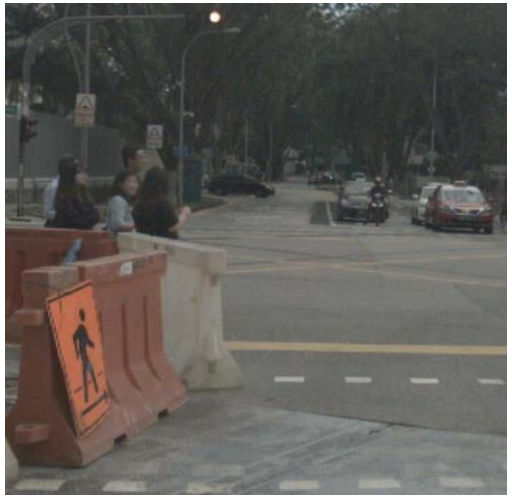
SCODE model

Zwicky export

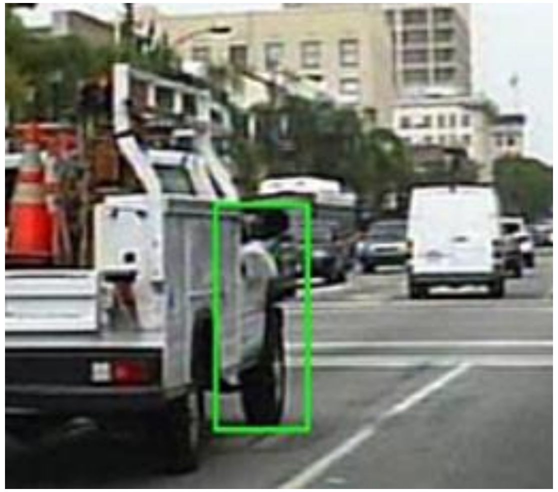
Pairwise combi-nations

Combi-natorial tool

Pairs, histogram, heuristics



FN VRU candidate, source: https://www.nuscenes.org/

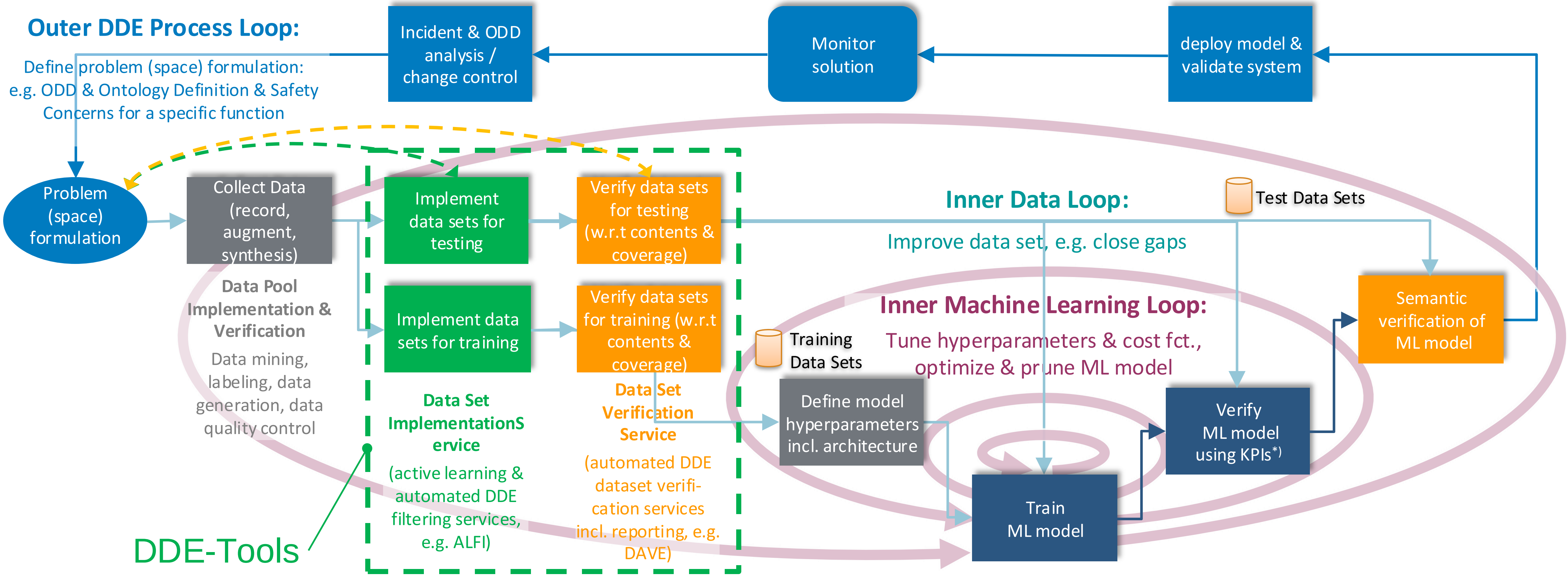


FP VRU candidate, source: https://www.frontiersin.org/articles/10.3389/fnbot.2018.00064/full

Data feature	Feature class combinations		
Longitudinal distance	Near	Short	Far
Lateral distance	Left lane	Ego lane	Right lane
Occlusion ratio	0…33%	33…66%	66…100%
Contrast	Low	Medium	High
Brightness	Dark	Medium	Bright

Systematically apply combinatorial methods to improve ODD coverage in important / safety-relevant data space regions.

What‘s in Machine Learning & Data-Driven Engineering Workflows



Outer DDE Process Loop:

Define problem (space) formulation: e.g. ODD & Ontology Definition & Safety Concerns for a specific function

Incident & ODD analysis / change control

Monitor solution

deploy model & validate system

Problem (space) formulation

Collect Data (record, augment, synthesis)

Data Pool Implementation & Verification

Data mining, labeling, data generation, data quality control

Implement data sets for testing

Verify data sets for testing (w.r.t contents & coverage)

Implement data sets for training

Verify data sets for training (w.r.t contents & coverage)

Data Set ImplementationService

(active learning & automated DDE filtering services, e.g. ALFI)

Data Set Verification Service

(automated DDE dataset verification services incl. reporting, e.g. DAVE)

DDE-Tools

Inner Data Loop:

Improve data set, e.g. close gaps

Test Data Sets

Semantic verification of ML model

Inner Machine Learning Loop:

Tune hyperparameters & cost fct., optimize & prune ML model

Training Data Sets

Define model hyperparameters incl. architecture

Train ML model

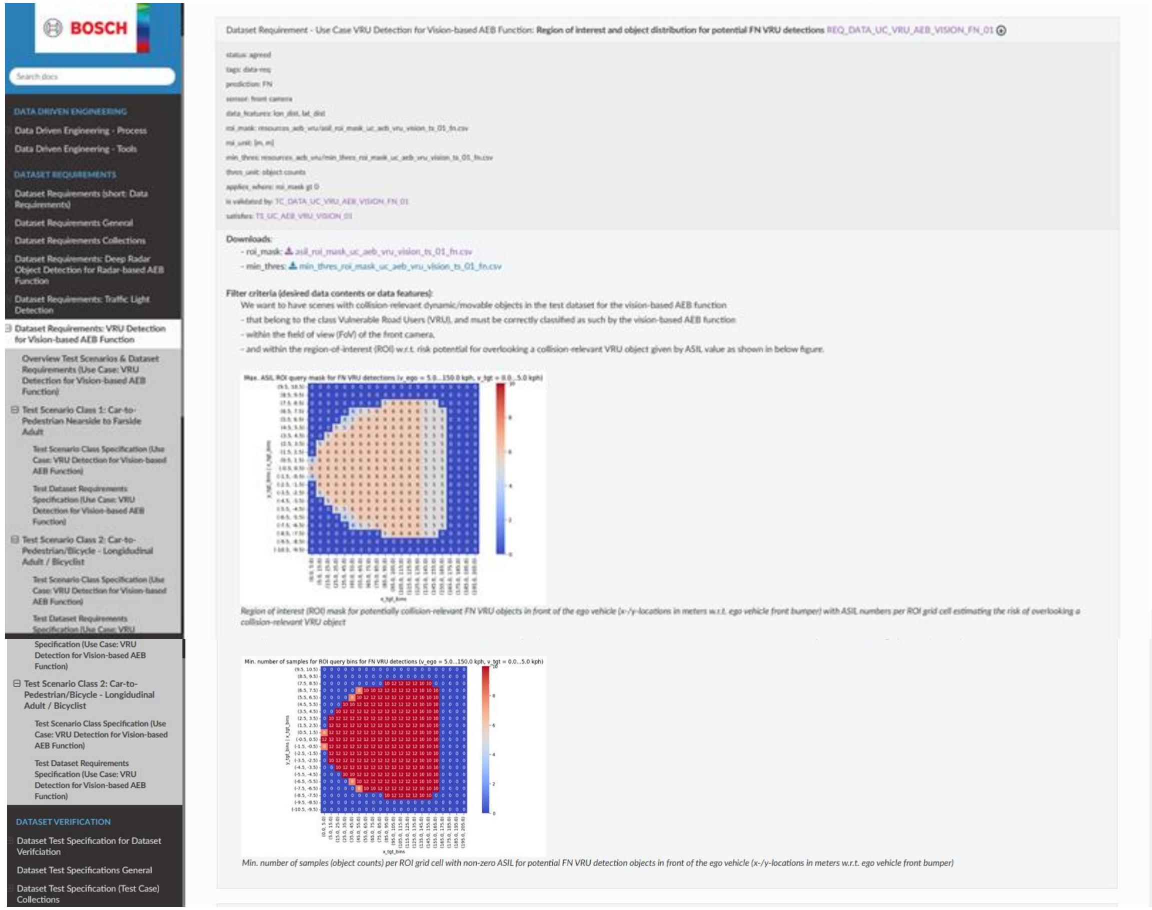
Verify ML model using KPIs[*)]

[*)] KPIs: metrics for certain characteristics applied to specifically compiled data sets

The DDE process closes an outer loop for systematic refinement of data requirements & development of verified ODD coverage.

AD: Automated Driving DDE: Data-Driven Engineering DNN: Deep Neural Network FoV: Field of View GT: Ground Truth ML: Machine Learning (connectionistic ML – esp. DNNs) ODD: Operational Design Domain TL: Traffic Light

Data Requirements on Safety-related Data Features or Properties



BOSCH

Search docs

DATA DRIVEN ENGINEERING

Data Driven Engineering - Process

Data Driven Engineering - Tools

DATASET REQUIREMENTS

Dataset Requirements (short: Data Requirements)

Dataset Requirements General

Dataset Requirements Collections

Dataset Requirements: Deep Radar Object Detection for Radar-based AEB Function

Dataset Requirements: Traffic Light Detection

⊟ Dataset Requirements: VRU Detection for Vision-based AEB Function

Overview Test Scenarios & Dataset Requirements (Use Case: VRU Detection for Vision-based AEB Function)

⊟ Test Scenario Class 1: Car-to-Pedestrian Nearside to Farside Adult

Test Scenario Class Specification (Use Case: VRU Detection for Vision-based AEB Function)

Test Dataset Requirements Specification (Use Case: VRU Detection for Vision-based AEB Function)

⊟ Test Scenario Class 2: Car-to-Pedestrian/Bicycle - Longidudinal Adult / Bicyclist

Test Scenario Class Specification (Use Case: VRU Detection for Vision-based AEB Function)

Test Dataset Requirements Specification (Use Case: VRU

Specification (Use Case: VRU Detection for Vision-based AEB Function)

⊟ Test Scenario Class 2: Car-to-Pedestrian/Bicycle - Longidudinal Adult / Bicyclist

Test Scenario Class Specification (Use Case: VRU Detection for Vision-based AEB Function)

Test Dataset Requirements Specification (Use Case: VRU Detection for Vision-based AEB Function)

DATASET VERIFICATION

Dataset Test Specification for Dataset Verifciation

Dataset Test Specifications General

Dataset Test Specification (Test Case) Collections

Dataset Requirement - Use Case VRU Detection for Vision-based AEB Function: **Region of interest and object distribution for potential FN VRU detections** REQ_DATA_UC_VRU_AEB_VISION_FN_01

status: agreed

tags: data-req

prediction: FN

sensor: front camera

data_features: lon_dist, lat_dist

roi_mask: resources_aeb_vru/asil_roi_mask_uc_aeb_vru_vision_ts_01_fn.csv

roi_unit: [m, m]

min_thres: resources_aeb_vru/min_thres_roi_mask_uc_aeb_vru_vision_ts_01_fn.csv

thres_unit: object counts

applies_where: roi_mask gt 0

is validated by: TC_DATA_UC_VRU_AEB_VISION_FN_01

satisfies: TS_UC_AEB_VRU_VISION_01

Downloads:

- roi_mask: asil_roi_mask_uc_aeb_vru_vision_ts_01_fn.csv
- min_thres: min_thres_roi_mask_uc_aeb_vru_vision_ts_01_fn.csv

Filter criteria (desired data contents or data features):

We want to have scenes with collision-relevant dynamic/movable objects in the test dataset for the vision-based AEB function

- that belong to the class Vulnerable Road Users (VRU), and must be correctly classified as such by the vision-based AEB function
- within the field of view (FoV) of the front camera,
- and within the region-of-interest (ROI) w.r.t. risk potential for overlooking a collision-relevant VRU object given by ASIL value as shown in below figure.

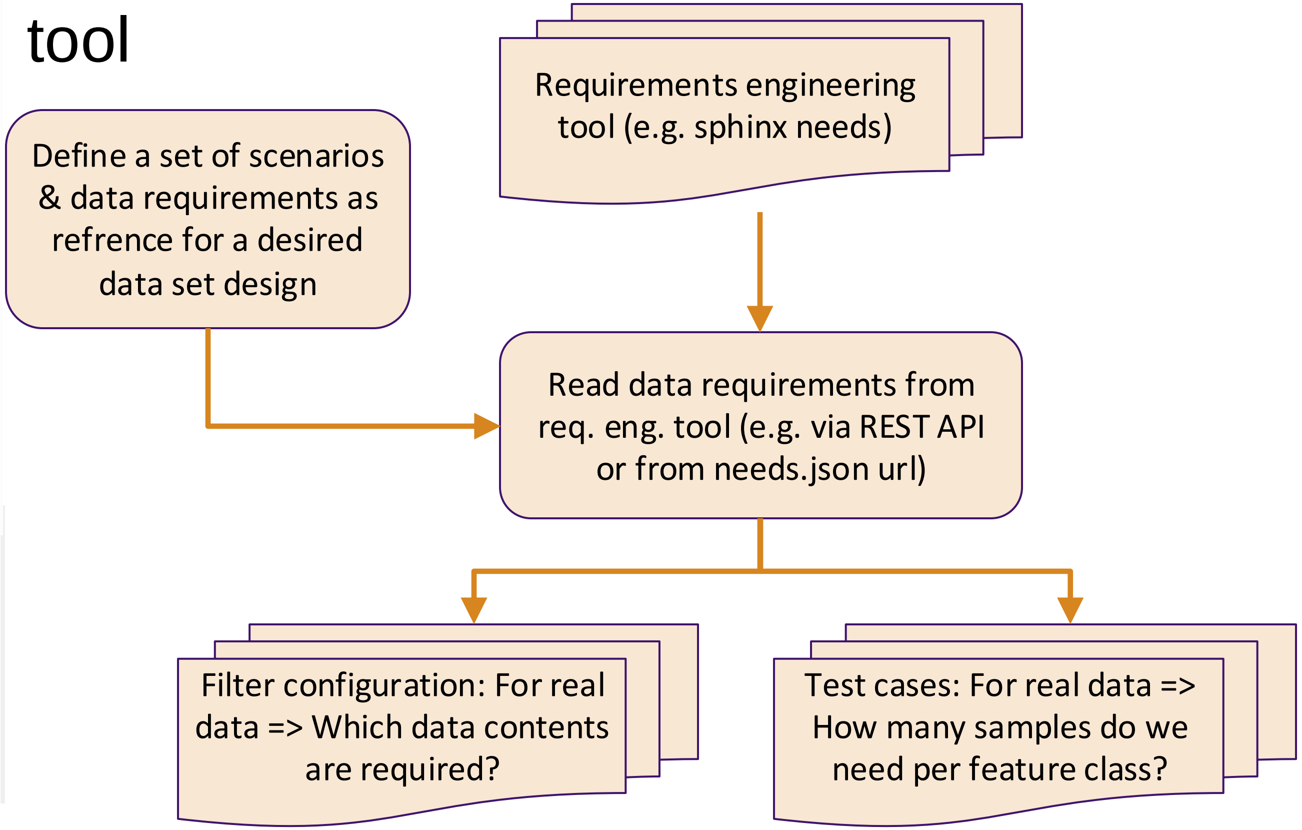
Max. ASIL ROI query mask for FN VRU detections (v_ego = 5.0...150.0 kph, v_tgt = 0.0...5.0 kph)
y_tgt_bins | x_tgt_bins
x_tgt_bins

Region of interest (ROI) mask for potentially collision-relevant FN VRU objects in front of the ego vehicle (x-/y-locations in meters w.r.t. ego vehicle front bumper) with ASIL numbers per ROI grid cell estimating the risk of overlooking a collision-relevant VRU object

Min. number of samples for ROI query bins for FN VRU detections (v_ego = 5.0...150.0 kph, v_tgt = 0.0...5.0 kph)
y_tgt_bins | x_tgt_bins
x_tgt_bins

Min. number of samples (object counts) per ROI grid cell with non-zero ASIL for potential FN VRU detection objects in front of the ego vehicle (x-/y-locations in meters w.r.t. ego vehicle front bumper)

Reading data requirements from req. eng. tool



Requirements engineering tool (e.g. sphinx needs)

Define a set of scenarios & data requirements as refrence for a desired data set design

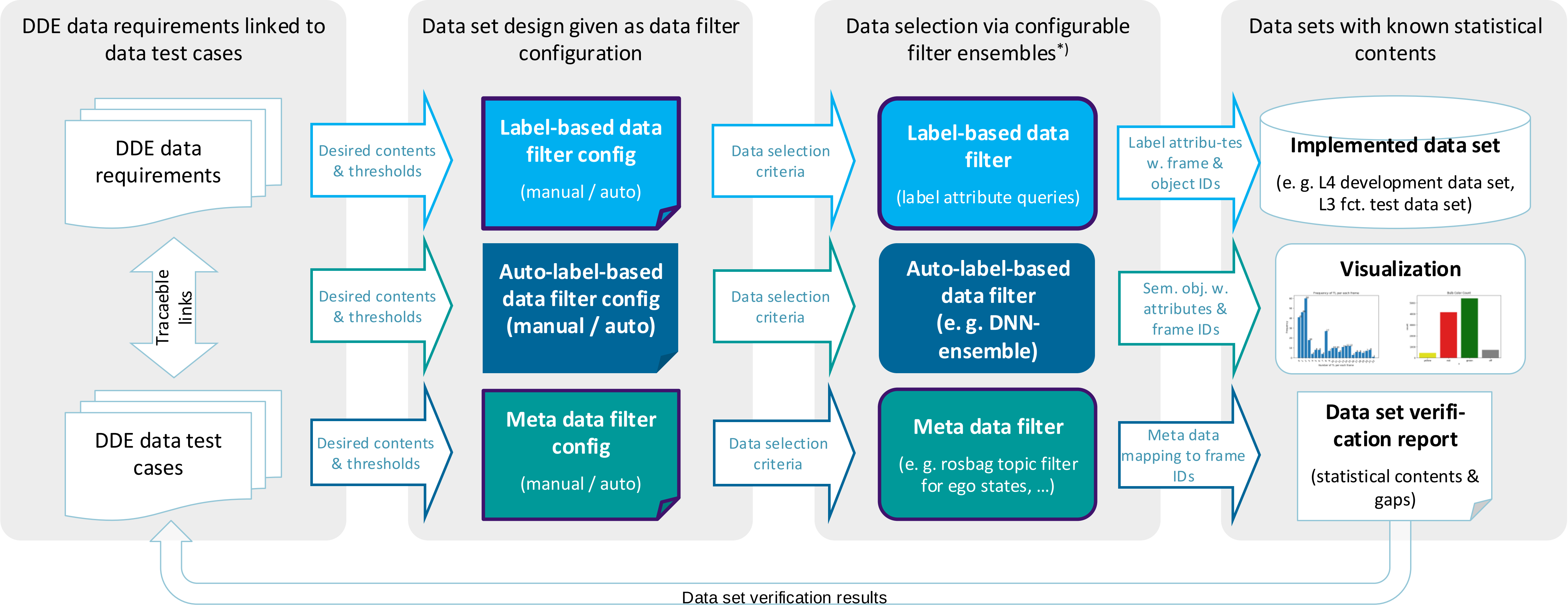
Read data requirements from req. eng. tool (e.g. via REST API or from needs.json url)

Filter configuration: For real data => Which data contents are required?

Test cases: For real data => How many samples do we need per feature class?

If we define data requirements in a machine-readable way we can automate the data set implementation & verification workflow.

Automated Data Set Implementation & Verification using Filter Ensembles



DDE data requirements linked to data test cases

DDE data requirements

Traceable links

DDE data test cases

Desired contents & thresholds

Desired contents & thresholds

Desired contents & thresholds

Data set design given as data filter configuration

Label-based data filter config
(manual / auto)

Auto-label-based data filter config (manual / auto)

Meta data filter config
(manual / auto)

Data selection criteria

Data selection criteria

Data selection criteria

Data selection via configurable filter ensembles*)

Label-based data filter
(label attribute queries)

Auto-label-based data filter (e. g. DNN-ensemble)

Meta data filter
(e. g. rosbag topic filter for ego states, …)

Label attribu-tes w. frame & object IDs

Sem. obj. w. attributes & frame IDs

Meta data mapping to frame IDs

Data sets with known statistical contents

Implemented data set
(e. g. L4 development data set, L3 fct. test data set)

Visualization

Data set verifi-cation report
(statistical contents & gaps)

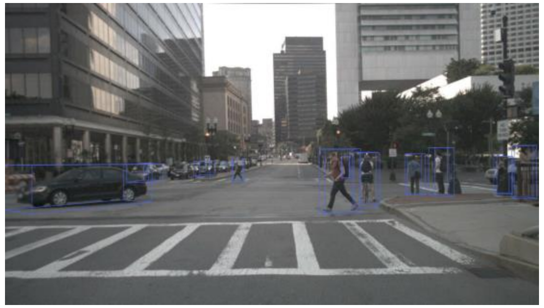
Data set verification results

Configurable auto-/label-based filters are key enablers to automatically compile data sets tailored to data requirements

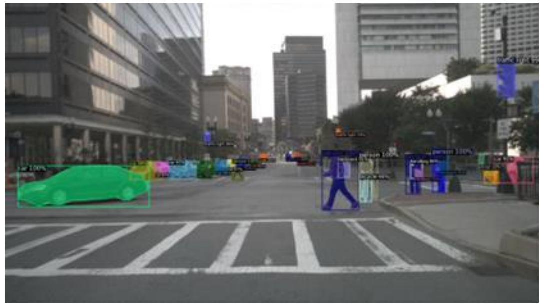
*) DDE data content filters are inspired from Business Values Assessment (BVA) ingest filter concept in Data Analytics Framework (DANF), but carry the idea further towards an autom. req.-based data set implementation & verification.

Available ALFI Task Modules (DNN Ensembles or Single Models)

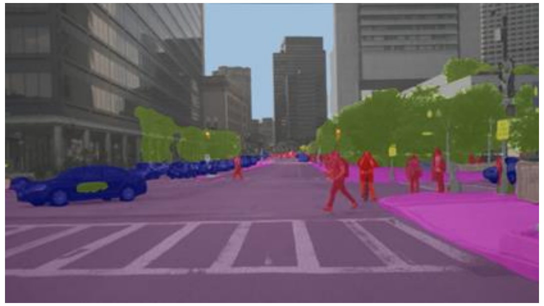
1. 3D object bounding boxes + distance & orientation



2./3. Instance segmentation mask + tight-fit 2D bboxes

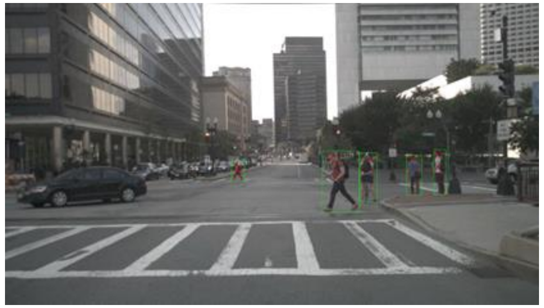


4. Semantic segmentation + free drivable space detection

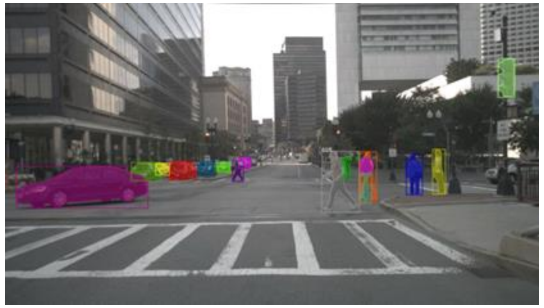


5. Depth map (relative depth)

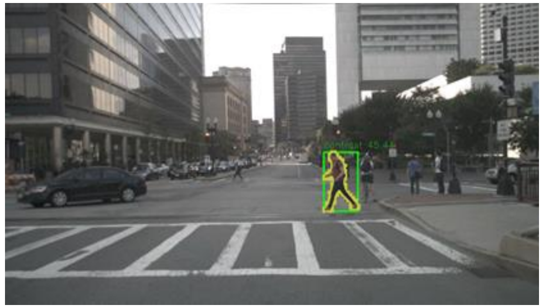
6. Skeleton keypoint estimation for VRU



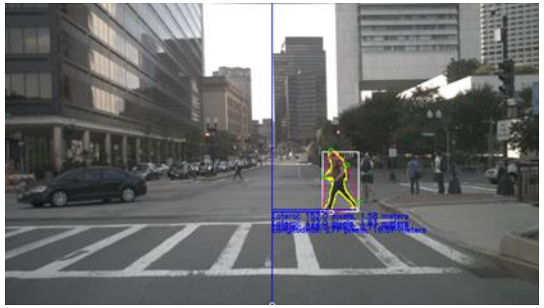
7. Occclusion detection (occluder & occludee + occlusion ratio)



8. Visual appearance meta data



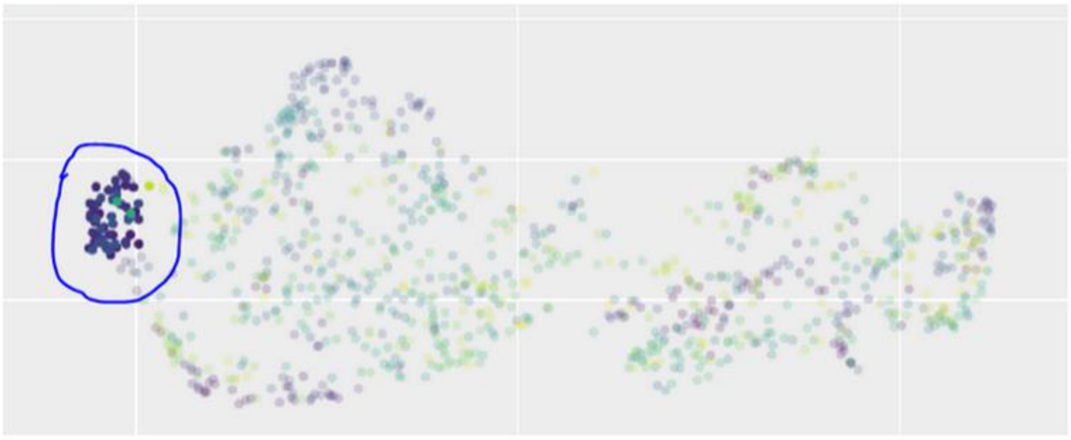
9. Mono camera distance estimation using calibration data (long./lat. dist.)



9 / 10 ALFI modules are available in dockerized form and partially as AML ensemble (truncation detection is not available yet)

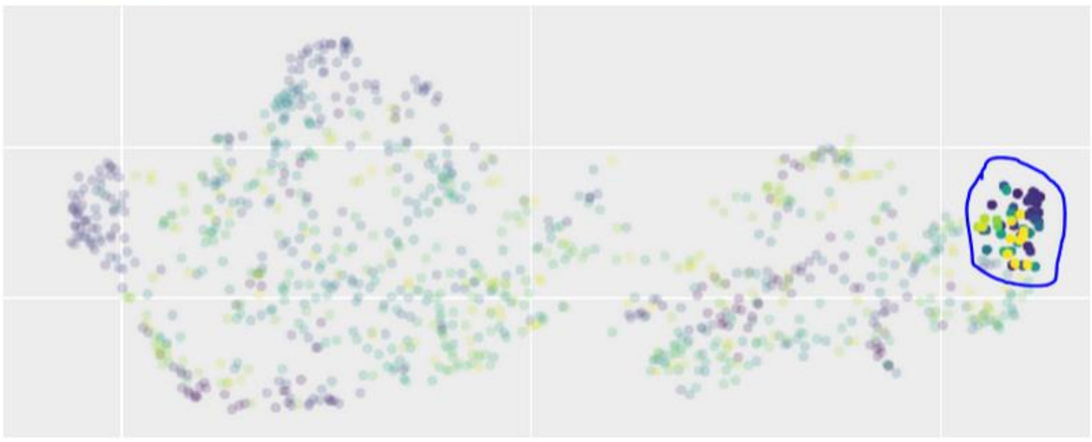
Embedded Feature Space Dataset Analysis using t-sne

next selected region:



images in selected region:

and one more region:



images inside selected region:

Using a combined embedded feature & semantic feature space analysis allows us to merge both human & model perspective.

Labeler Apps

- Label ground truth for image, video, lidar and medical data

- Important for training networks for:
  - Classifiers
  - Object Detectors
  - Segmentation

- Features:
  - Create label definitions and attributes.
  - Semi automated or automated labeling with built-in or custom algorithms
  - Blocked processing support (image)
  - Superpixel automation (Image, Video)

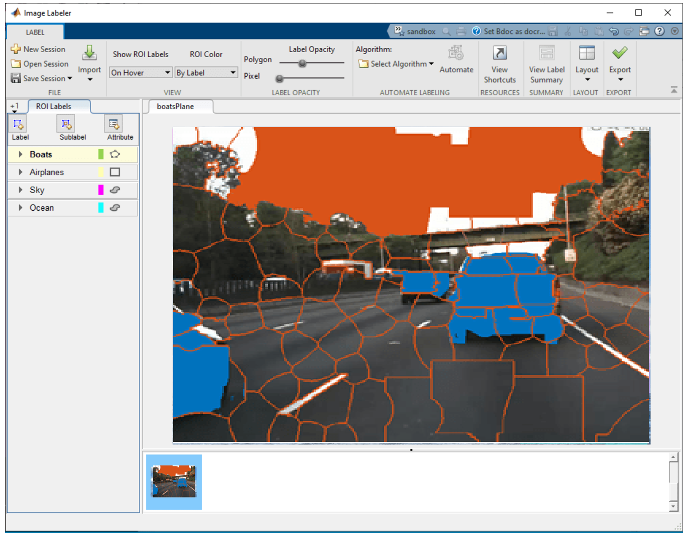
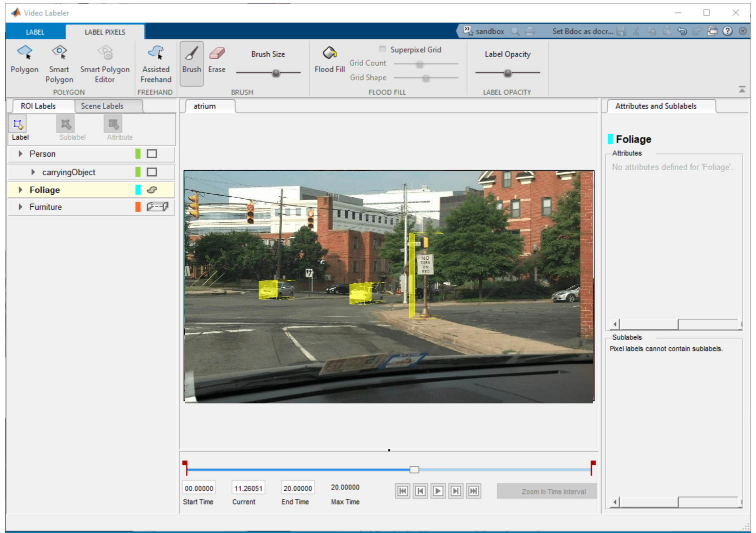
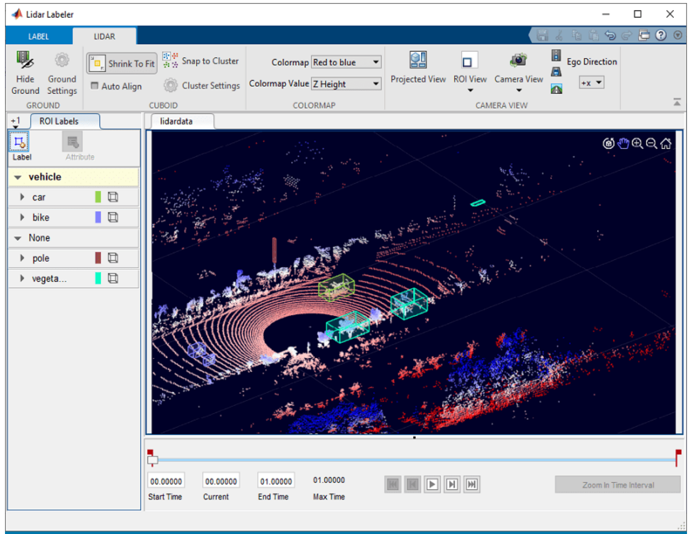


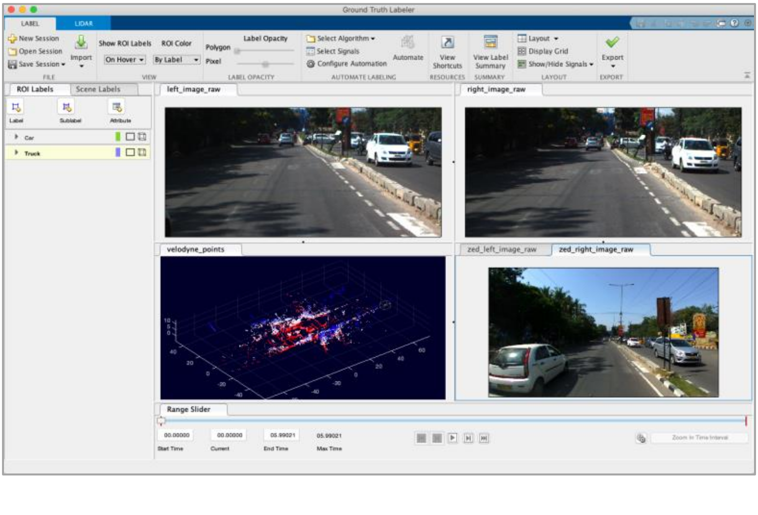
Image Labeler (Computer Vision Toolbox)



Video Labeler (Computer Vision Toolbox)



Lidar Labeler (Lidar Toolbox)



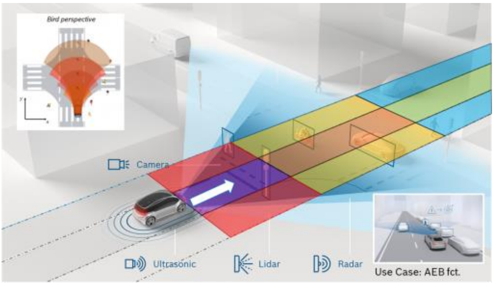
Ground Truth Labeler (Automated Driving Toolbox)

- **Convert Image Labeler ground truth data to OpenLABEL Format**

Example how to Design & Verify AI Test Datasets using DDE

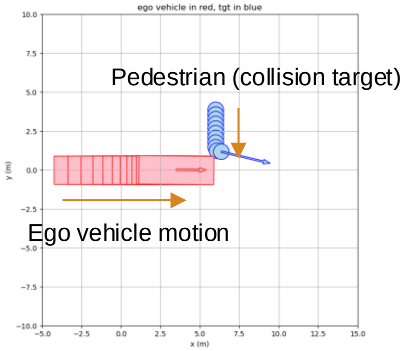
System & ODD & UC analysis

Test scenario catalogue



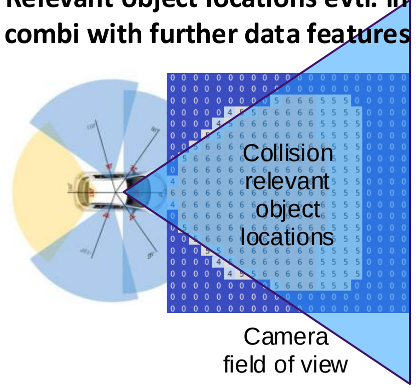
Hazard & risk analysis

Collision simulation



Requirements elicitation

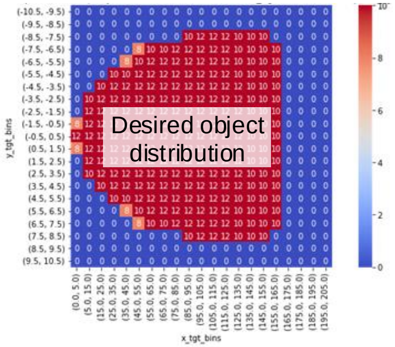
Relevant object locations evtl. in combi with further data features



AI requirements elicitation

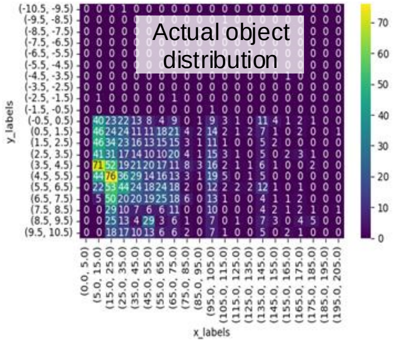
AI dataset requirements

Region of interest & desired object distribution (BEV)



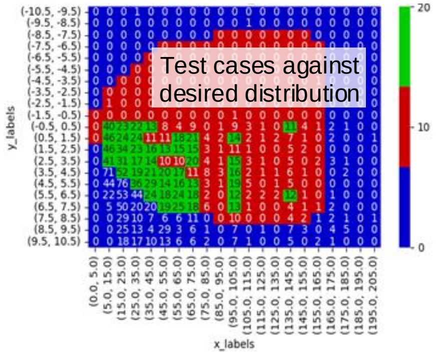
Query against auto-/label database

Dataset implementation by requirement-based filtering



Desired vs. actual distribut.

Datase verification incl. coverage & gap analysis



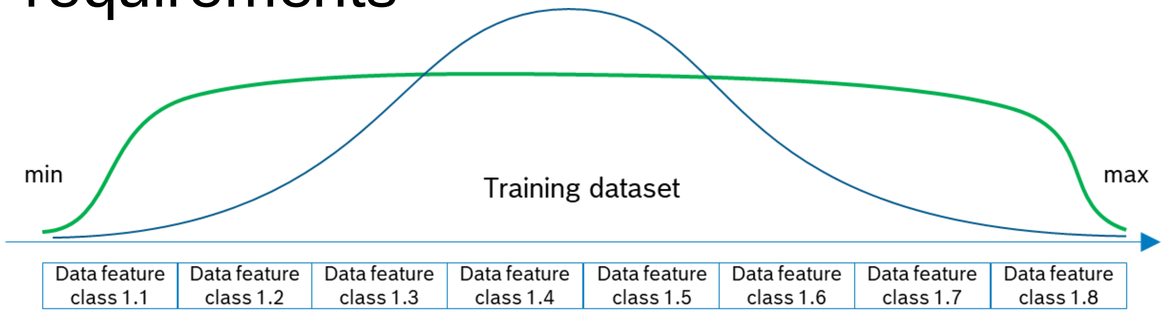
AI perform. eval. vs. data features

We need quality control for our AI datasets w.r.t. data coverage, contents, diversity, balance & integrity – not only label quality!

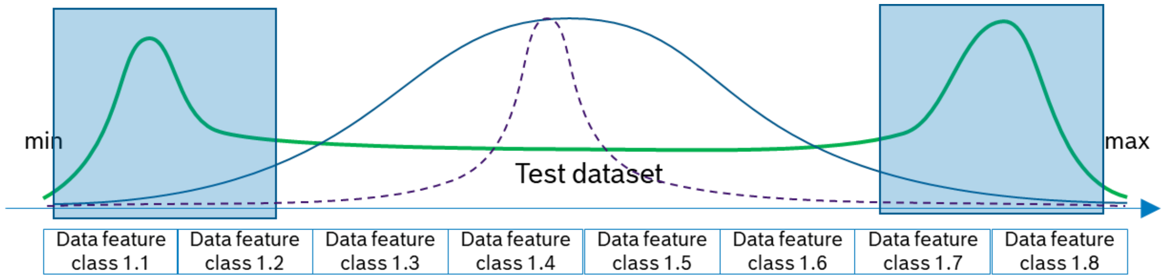
AI: Artificial Intelligence BEV: Bird's eye view DDE: Data Driven Engineering FoV: Field of View GT: Ground Truth ML: Machine Learning ODD: Operational Design Domain ROI: Region-of-interest

Goal: Tailor Datasets Content-wise against Dataset Requirements

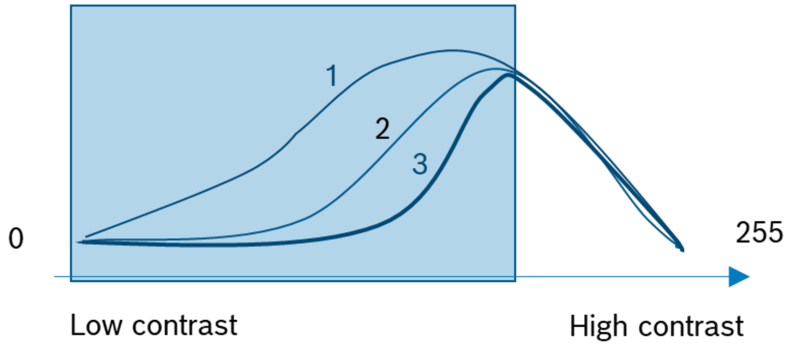
Tailor training datasets against data requirements



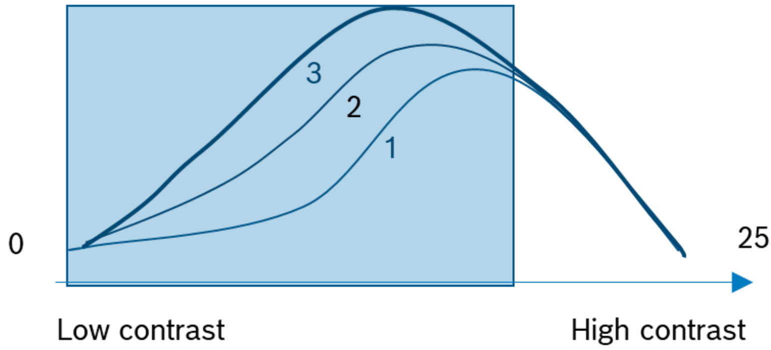
Tailor test datasets against data requirements



Reduce training / test data coverage in certain regions

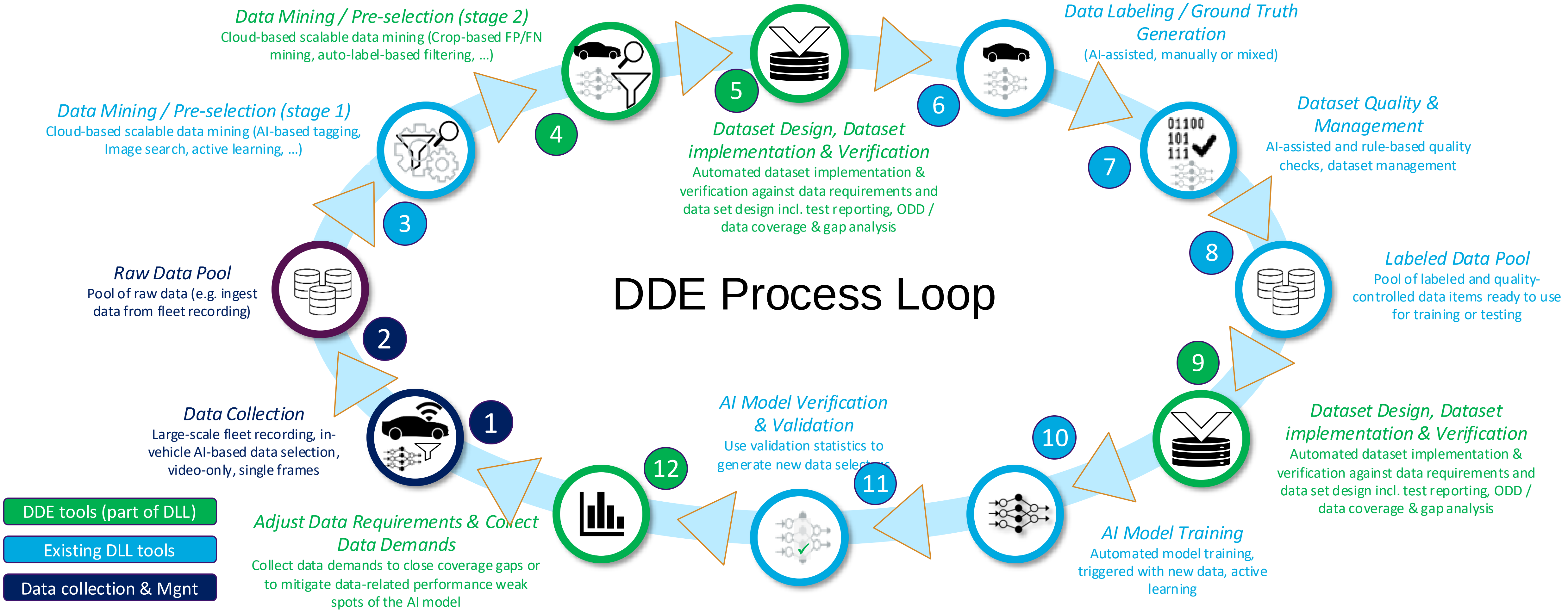


Increase training / test data coverage in certain regions



DDE-tools like ALFI & DAVE allow tailoring & verification of real datasets w.r.t. contents/features against dataset requirements.

Interactions of DDE Process & DLL Tools along the Process Flow



DDE Process Loop

1 *Data Collection*
Large-scale fleet recording, in-vehicle AI-based data selection, video-only, single frames

2 *Raw Data Pool*
Pool of raw data (e.g. ingest data from fleet recording)

3 *Data Mining / Pre-selection (stage 1)*
Cloud-based scalable data mining (AI-based tagging, Image search, active learning, …)

4 *Data Mining / Pre-selection (stage 2)*
Cloud-based scalable data mining (Crop-based FP/FN mining, auto-label-based filtering, …)

5 *Dataset Design, Dataset implementation & Verification*
Automated dataset implementation & verification against data requirements and data set design incl. test reporting, ODD / data coverage & gap analysis

6 *Data Labeling / Ground Truth Generation*
(AI-assisted, manually or mixed)

7 *Dataset Quality & Management*
AI-assisted and rule-based quality checks, dataset management

8 *Labeled Data Pool*
Pool of labeled and quality-controlled data items ready to use for training or testing

9 *Dataset Design, Dataset implementation & Verification*
Automated dataset implementation & verification against data requirements and data set design incl. test reporting, ODD / data coverage & gap analysis

10 *AI Model Training*
Automated model training, triggered with new data, active learning

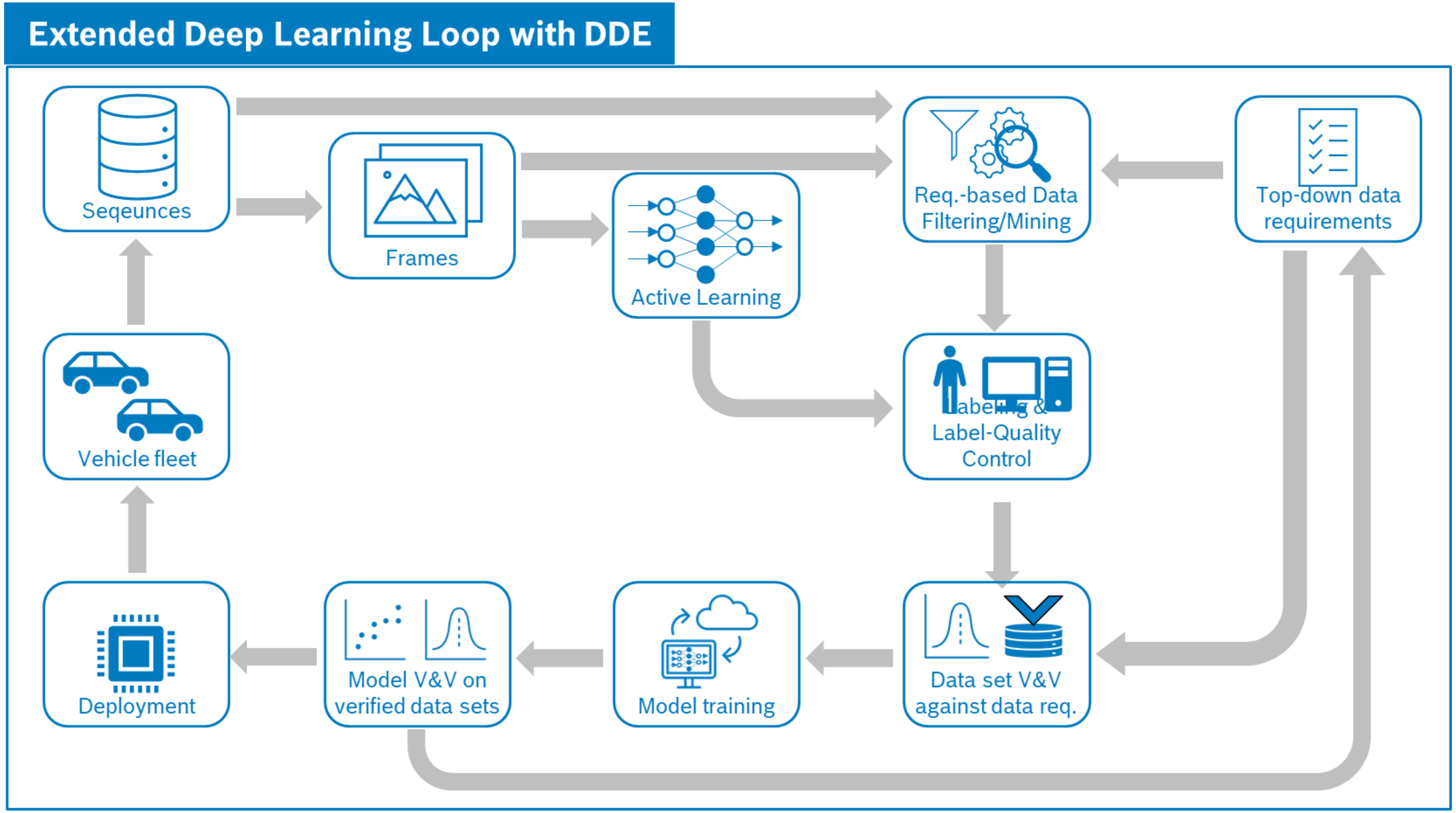
11 *AI Model Verification & Validation*
Use validation statistics to generate new data selectors

12 *Adjust Data Requirements & Collect Data Demands*
Collect data demands to close coverage gaps or to mitigate data-related performance weak spots of the AI model

- DDE tools (part of DLL)
- Existing DLL tools
- Data collection & Mgnt

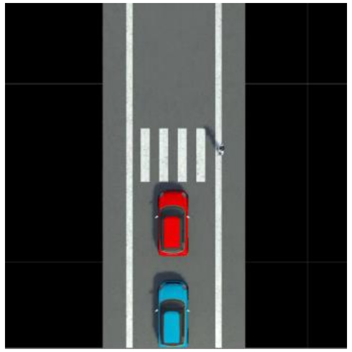
A mini DDE loop has been closed including the steps 3, 4, 5, 9, 10, 11, 12, which we can operate to execute training experiments.

Combine Data-Driven Engineering & Active Learning synergetically


Extended Deep Learning Loop with DDE
Seqeunces
Frames
Active Learning
Req.-based Data Filtering/Mining
Top-down data requirements
Vehicle fleet
Labeling & Label-Quality Control
Deployment
Model V&V on verified data sets
Model training
Data set V&V against data req.

Data-Driven Engineering (top-down view) & Active Learning (bottom-up view) complement one another in a deep learning loop.

Outlook: Scenario Class Coverage given a Scenario Catalogue

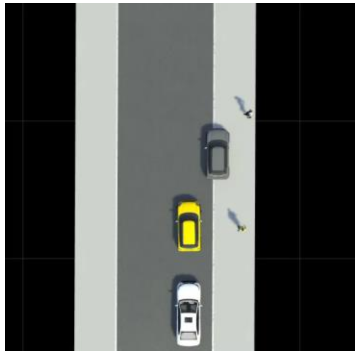


Scenario ID 00X-000

Class name: …
Config file: …
Description: …
Parameters: …
Data Coverage: …

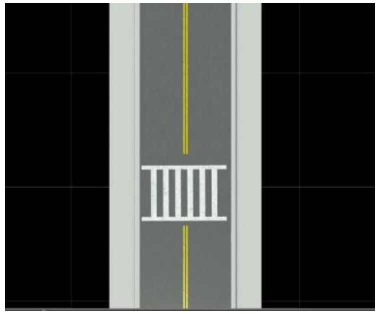
Scenario ID 00X-001

Class name: …
Config file: …
Description: …
Parameters: …



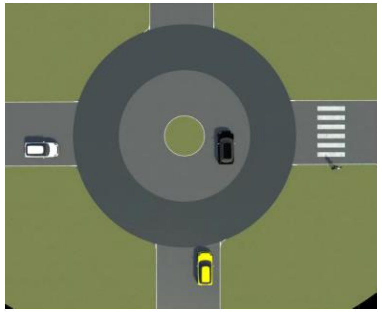
Scenario ID 00X-002

Class name: …
Config file: …
Description: …
Parameters: …



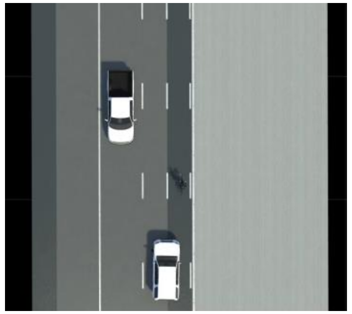
Scenario ID 00X-003

Class name: …
Config file: …
Description: …
Parameters: …



Scenario ID 00X-004

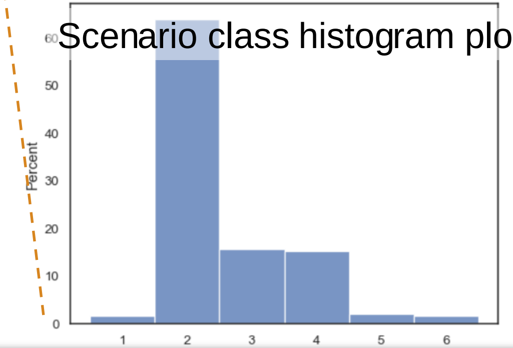
Class name: …
Config file: …
Description: …
Parameters: …



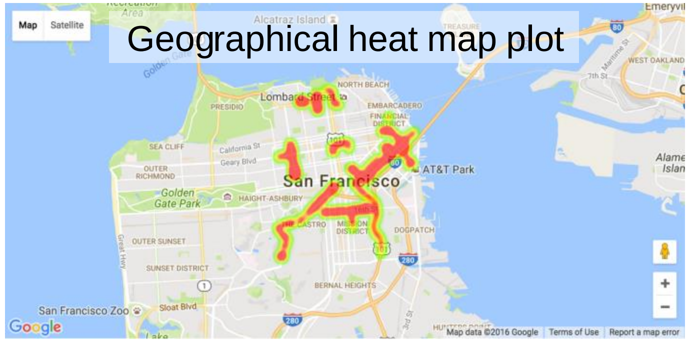
Scenario ID 00X-005

Class name: …
Config file: …
Description: …
Parameters: …

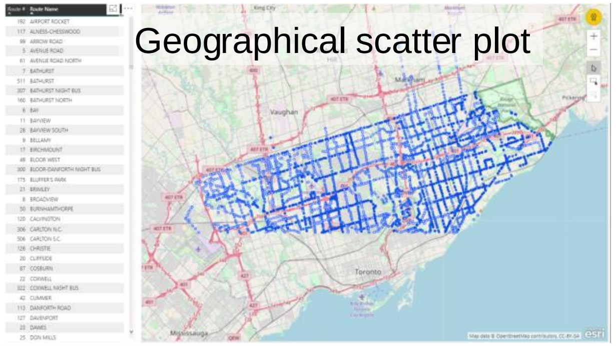
…



Scenario class histogram plot



Geographical heat map plot



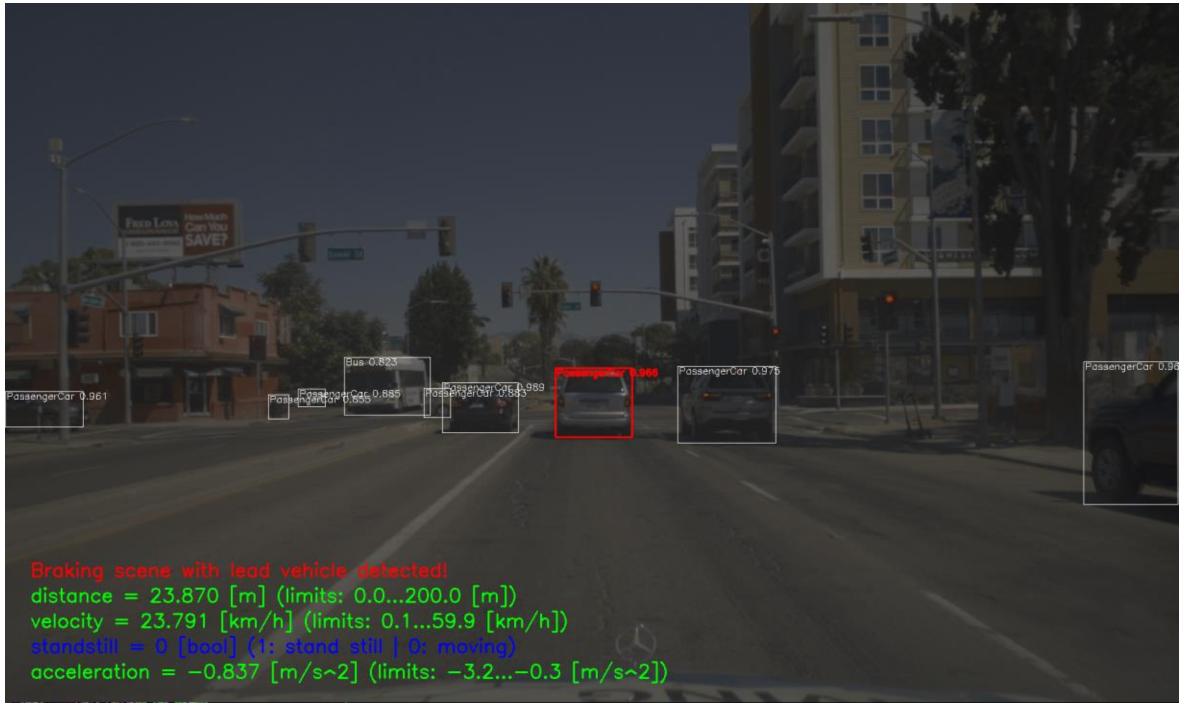
Geographical scatter plot

…

We cut our Operational Design Domain (ODD) into a collection of abstract scenario classes for which we doe a safety analysis.

Outlook: Classify Scenarios using Sequential Multi-Sensor Data

Framewise detect braking scenes with lead vehicle

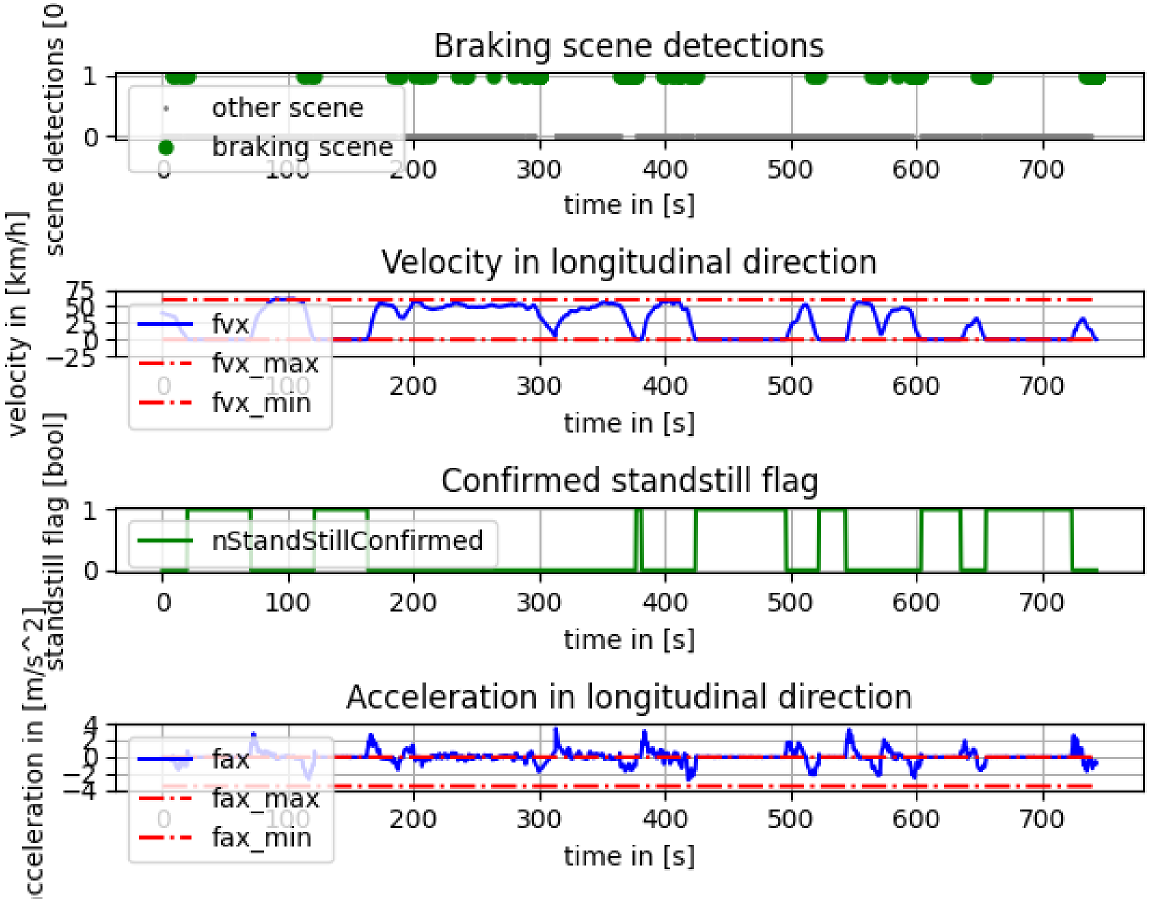


Braking scene with lead vehicle detected!
distance = 23.870 [m] (limits: 0.0...200.0 [m])
velocity = 23.791 [km/h] (limits: 0.1...59.9 [km/h])
standstill = 0 [bool] (1: stand still | 0: moving)
acceleration = −0.837 [m/s~2] (limits: −3.2...−0.3 [m/s~2])

Configurable filter parameters:

- Object classes, lane bins, distance bins w. tolerance, driving state min/max thresholds with tolerance

Detect braking scenes with lead vehicle over time



For scenario identification we need to look into (multi-sensor) sequential data considering spatio-temporal effects.

2024 MathWorks
中国汽车年会

Thank you

MathWorks