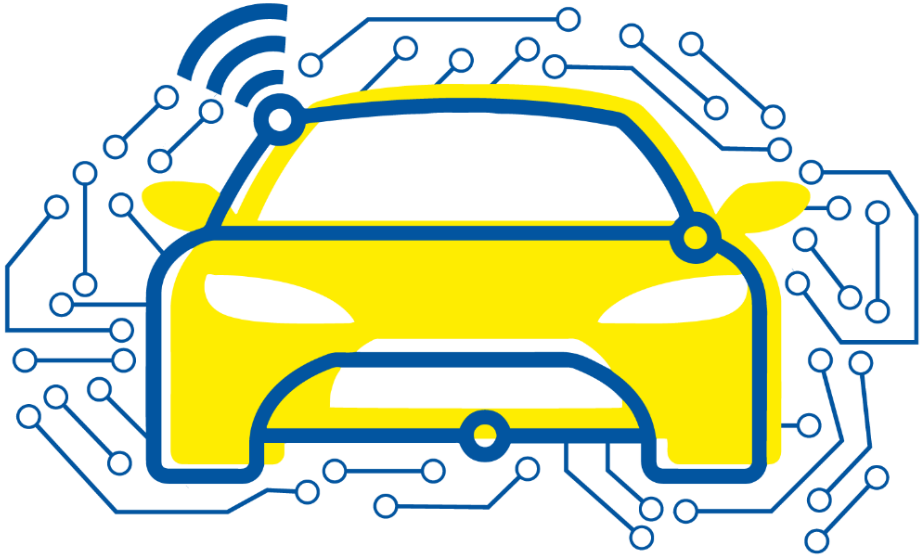
Automated and Connected Driving Challenges

Section 2 – Sensor Data Processing

Point Cloud Occupancy Grid Mapping

Training & Evaluation



Bastian Lampe

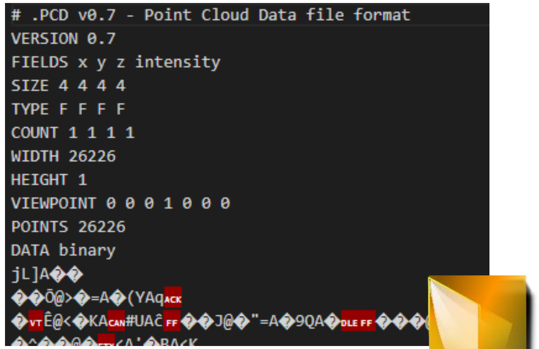
Institute for Automotive Engineering

Point Cloud OGM – Training & Evaluation

Data Pipeline

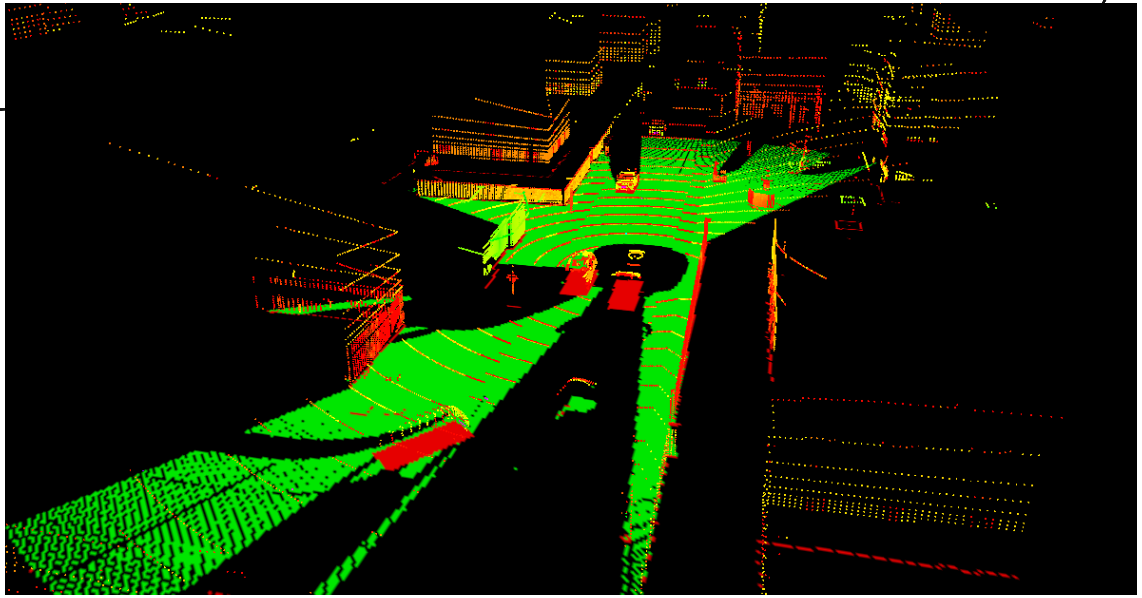
- How to **store** the training data?
  - Common file types (e.g. PCD and PNG) vs. dedicated data structure (e.g. TFRecord)
- How to **load** the training data into the deep learning framework?

Point Cloud



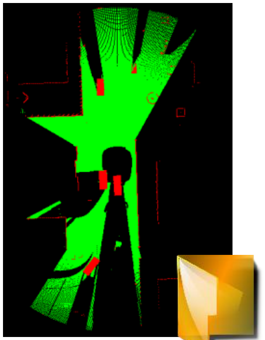
```
# .PCD v0.7 - Point Cloud Data file format
VERSION 0.7
FIELDS x y z intensity
SIZE 4 4 4 4
TYPE F F F F
COUNT 1 1 1 1
WIDTH 26226
HEIGHT 1
VIEWPOINT 0 0 0 1 0 0 0
POINTS 26226
DATA binary
jL]A��
��Ō@>�=A�(YAqACK
�VTÊ@<�KACAN#UAĉFF��J@�"=A�9QA�DLE FF���(
�^���@�STX<A'�BA<K
```

input_01.pcd



Simulated Input Point Cloud and Label Grid Map

Grid Map



label_01.png

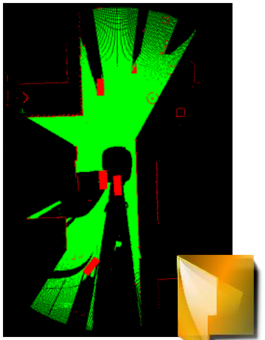
Images: ika

Point Cloud OGM – Training & Evaluation

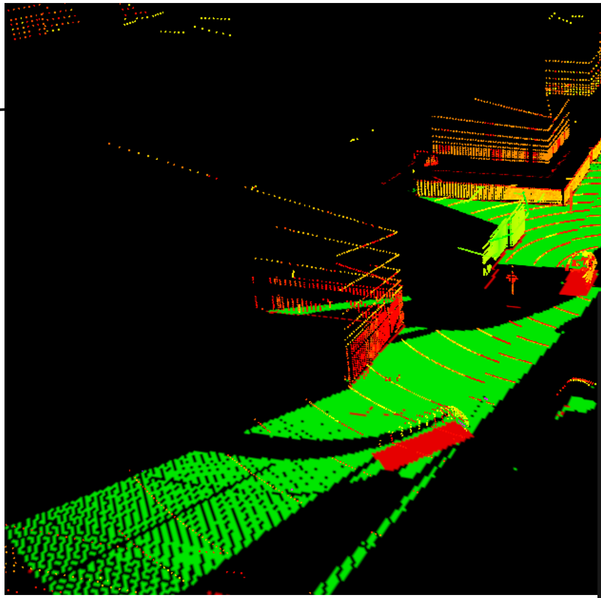
Data Pipeline

- How to **store** the training data?
  - Common file types (e.g. PCD and PNG) vs. dedicated data structure (e.g. TFRecord)
- How to **load** the training data into the deep learning framework?
  - Loading native files from disk vs. using TensorFlow Datasets API

Grid Map

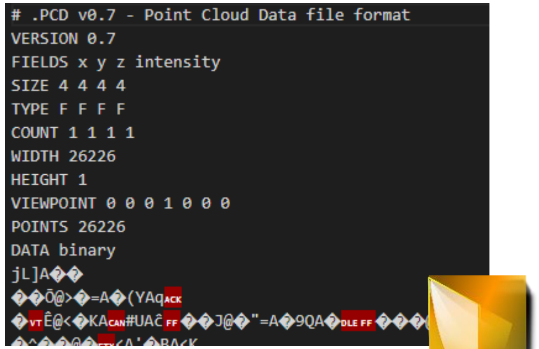


label_01.png



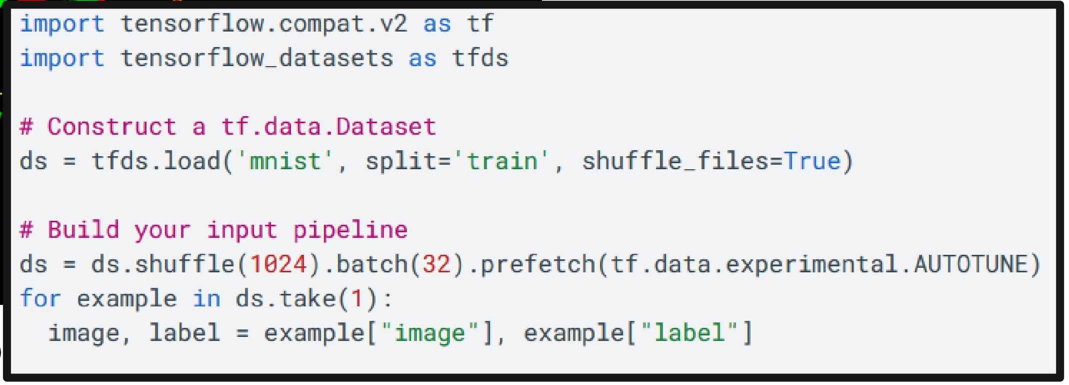
Simulated Input Point Clo

Point Cloud



```
# .PCD v0.7 - Point Cloud Data file format
VERSION 0.7
FIELDS x y z intensity
SIZE 4 4 4 4
TYPE F F F F
COUNT 1 1 1 1
WIDTH 26226
HEIGHT 1
VIEWPOINT 0 0 0 1 0 0 0
POINTS 26226
DATA binary
jL]A��
��Ō@>�=A�(YAqACK
�VTÊ@<�KACAN#UAČFF��J@�"=A�9QA�DLE FF���
```

input_01.pcd



```python
import tensorflow.compat.v2 as tf
import tensorflow_datasets as tfds

# Construct a tf.data.Dataset
ds = tfds.load('mnist', split='train', shuffle_files=True)

# Build your input pipeline
ds = ds.shuffle(1024).batch(32).prefetch(tf.data.experimental.AUTOTUNE)
for example in ds.take(1):
  image, label = example["image"], example["label"]
```

https://www.tensorflow.org/datasets

Images: ika

Point Cloud OGM – Training & Evaluation

Evaluation Metrics

- How to **measure** the performance of the trained model?
  - Binary evaluation for states „Free“ and „Occupied“
    - Precision $= \frac{\mathrm{TP}}{\mathrm{TP+FP}}$ (Positive Predictive Value)
    - Recall $= \frac{\mathrm{TP}}{\mathrm{TP+FN}}$ (True Positive Rate)
    - Thresholds Θ_O/Θ_F for belief masses

b_O	belief mass for the cell being **occupied**
b_F	belief mass for the cell being **free**
u	**uncertainty** mass

$$b_O + b_F + u = 1$$

$$b_O > \Theta_O \rightarrow Occupied$$
$$b_F > \Theta_F \rightarrow Free$$

		actual	
		free	not free
predicted	free	True Positives (TP)	False Positives (FP)
	not free	False Negatives (FN)	True Negatives (TN)

State “Free”

		actual	
		occupied	not occupied
predicted	occupied	True Positives (TP)	False Positives (FP)
	not occupied	False Negatives (FN)	True Negatives (TN)

State “Occupied”

Point Cloud OGM – Training & Evaluation

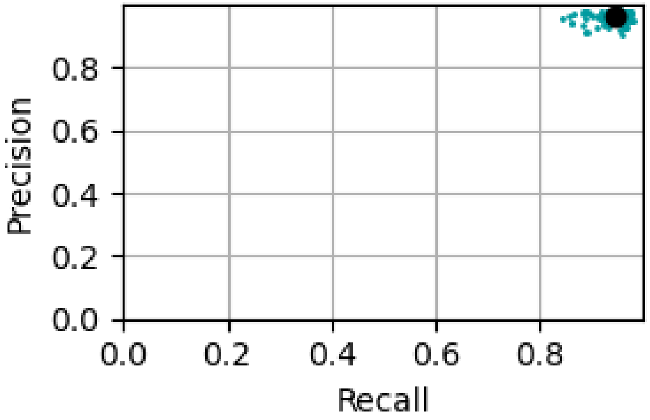
Evaluation Metrics

- How to **measure** the performance of the trained model?
  - Binary evaluation for states „Free“ and „Occupied“
    - Precision $= \frac{\mathrm{TP}}{\mathrm{TP+FP}}$ (Positive Predictive Value)
    - Recall $= \frac{\mathrm{TP}}{\mathrm{TP+FN}}$ (True Positive Rate)
    - Thresholds Θ_O/Θ_F for belief masses

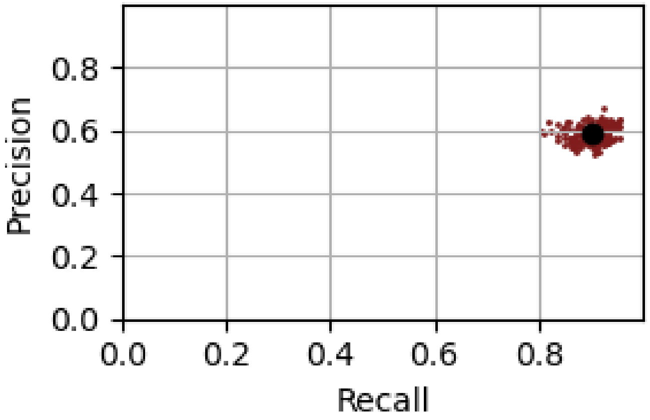
b_O	belief mass for the cell being **occupied**
b_F	belief mass for the cell being **free**
u	**uncertainty** mass

$$b_O + b_F + u = 1$$

$$b_O > \Theta_O \rightarrow Occupied$$
$$b_F > \Theta_F \rightarrow Free$$



State “Free”

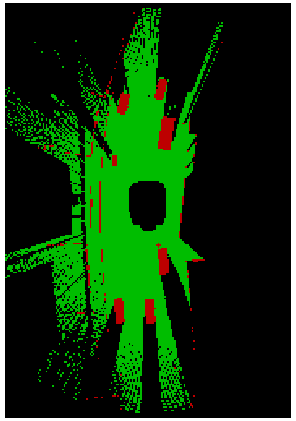


State “Occupied”

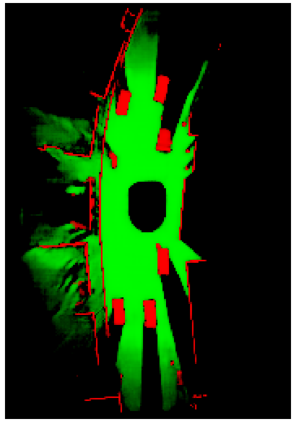
Point Cloud OGM – Training & Evaluation

Evaluation Metrics

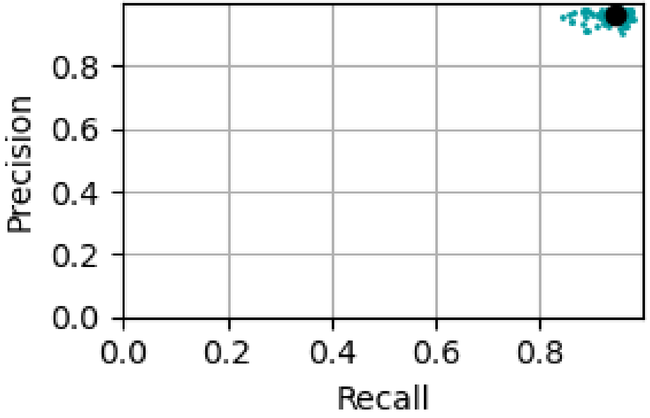
- How to **measure** the performance of the trained model?
  - Binary evaluation for states „Free“ and „Occupied“
    - Precision $= \frac{\text{TP}}{\text{TP+FP}}$ (Positive Predictive Value)
    - Recall $= \frac{\text{TP}}{\text{TP+FN}}$ (True Positive Rate)
    - Thresholds Θ_O/Θ_F for belief masses



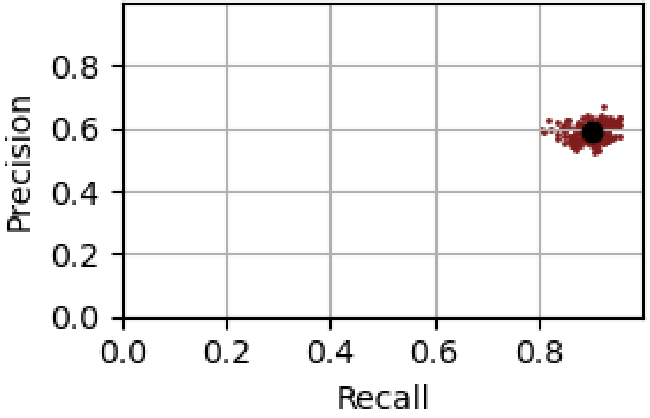
Label



Prediction



State “Free”



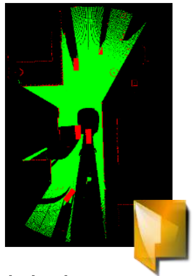
State “Occupied”

Images: ika

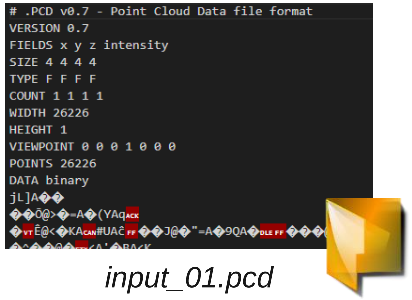
Point Cloud OGM – Training & Evaluation

Summary

- An **efficient data pipeline** can speed up the training substantially



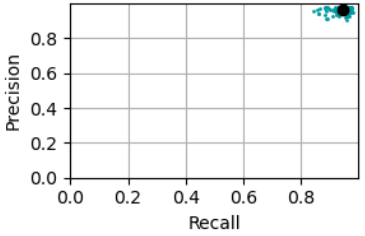
label_01.png

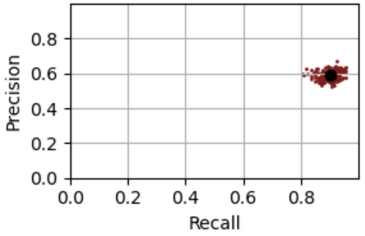


```
# .PCD v0.7 - Point Cloud Data file format
VERSION 0.7
FIELDS x y z intensity
SIZE 4 4 4 4
TYPE F F F F
COUNT 1 1 1 1
WIDTH 26226
HEIGHT 1
VIEWPOINT 0 0 0 1 0 0 0
POINTS 26226
DATA binary
jL]A��
��Ō@>�=A�(YAqACK
�VTÊ@<�KACAN#UAĉFF��J@�"=A�9QA�DLE FF���
```

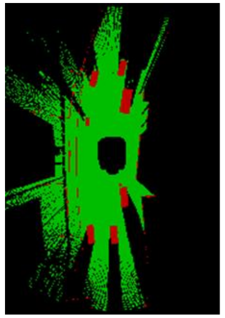
input_01.pcd

- **Different metrics** evaluate different performance aspects

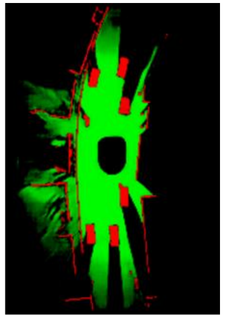




- **Analyze and understand** the model errors



Label



Prediction

Images: ika