Karlsruhe Institute of Technology

Benchmarking Anomaly Detection on Camera and Lidar Data with 3D Voxel Representation

Bachelor Thesis

Lukas Namgyu Roessler

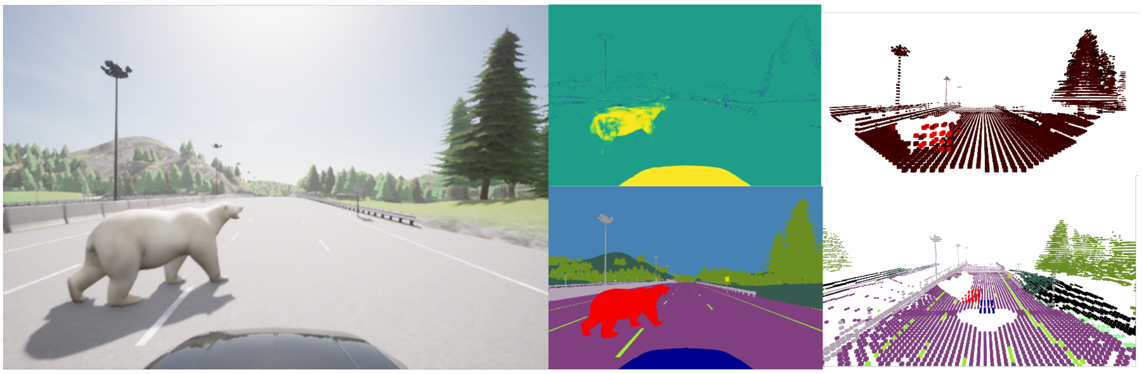
Department of Economics and Management
Institute of Applied Informatics and Formal Description Methods
and
FZI Research Center for Information Technology

Reviewer:	Prof. Dr.–Ing. J. M. Zöllner
Second reviewer:	Prof. Dr. A. Oberweis
Advisor:	M.Sc. Daniel Bogdoll

Research Period: 1. June 2023 – 30. November 2023

Benchmarking Anomaly Detection on Camera and Lidar Data with 3D Voxel Representation

by
Lukas Namgyu Roessler



Bachelor Thesis
November 2023

Bachelor Thesis, FZI
Department of Economics and Management, 2023
Reviewers: Prof. Dr.–Ing. J. M. Zöllner, Prof. Dr. A. Oberweis

Abteilung Technisch Kognitive Assistenzsysteme
FZI Research Center for Information Technology

Affirmation

Ich versichere wahrheitsgemäß, die Arbeit selbstständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde.

Karlsruhe,
November 2023

Lukas Namgyu Roessler

Abstract

The research field of autonomous driving has seen rapid development in recent years. There are, however, challenges that hinder the deployment of autonomous vehicles on the road. One of these challenges is the detection of unknown or anomalous instances on the road. The field of Anomaly Detection is crucial for the safe deployment of these systems, as detection failure could lead to the execution of potentially dangerous behavior.

Most autonomous vehicles employ an array of different sensors for scene understanding. To effectively utilize data extracted from multiple sensors, it is important to fuse all sensor data into a common state representation.

This thesis explores anomaly detection in combination with sensor fusion representations by evaluating anomaly detection methods for camera and LiDAR on voxel grids. The current state-of-the-art of anomaly detection for camera and LiDAR is reviewed to identify current trends and research gaps in the field. From the literature review, a camera-based method and a LiDAR-based method were selected for evaluation on the FZI AnoVox benchmark, an anomaly detection dataset that includes ground truth information on 3D surroundings in the form of a voxel grid. Anomaly score predictions were mapped into the voxel space to evaluate the detection performance. The findings show that metric results on images and point clouds can significantly deviate when transformed into voxels. Furthermore, the thesis explores effects and result changes when adjusting parameters in the voxelization process.

Contents

1 Introduction

The research field of Autonomous Driving has seen significant progress with the introduction of Deep Learning[40]. Despite the remarkable advances in recent years, autonomous vehicles are still not yet commonly deployed on the road. One of the main reasons for this is their weakness in handling unexpected or abnormal scenarios. Failure to correctly detect an anomaly can lead to dangerous situations as the vehicle may attempt to execute a maneuver that could potentially endanger its passengers or other traffic participants. The research field of anomaly detection tries to tackle this problem by developing methods that can effectively detect such anomalies on the road. These methods are specialized in detecting unknown events and are set out to handle unexpected scenarios. This work focuses on anomalies that Breitenstein et al.[13] refer to as "Anomalies on the Object Level", which encompasses the detection of objects that a vehicle is unlikely to encounter on the road.

Traditionally, autonomous vehicles are equipped with multiple types of sensors in addition to cameras, like LiDAR sensors. LiDAR (Light Detection and Ranging) sensors send out laser impulses and calculate the distance traveled by measuring the time difference between the initial emission and the reception of the impulses' reflection. A perception system would then receive the sensory data in the shape of a 3D point cloud with precise distance measurements for each point. Multimodal sensor setups have the great advantage of integrating the strengths of each sensor into a common scene understanding. To give an example, a LiDAR sensor can complement a camera very well due to its accurate depth measurement whilst the camera can capture features of surrounding instances more detailed than a LiDAR scan. To effectively combine the information provided by all sensors, each sensor's information has to be combined into a unified representation where the vehicle can predict and calculate trajectories.

One such common state representation for unifying different sensor data is a voxel space. Voxels can be understood as cubic elements in 3D space and offer numerous properties advantageous for real-time computation in autonomous driving[4].

Presently, most research in anomaly detection in the context of autonomous driving focuses on image-based detection[11, 12]. As an autonomous vehicle with multimodal sensor equipment will, however, transform its sensor representation into the sensor fusion representation, it is of adequate interest how anomaly detection methods for individual sensors perform there.

The objective of this thesis is to compare anomaly detection for camera sensors and anomaly detection for LiDAR sensors in a voxel space and analyze their performance when transforming results into a voxel volume. To allow a fair comparison of both sensor domains, state-of-the-art anomaly detection methods were chosen for both camera and LiDAR. Retraining both methods on the same training data furthermore ensures that the methods have the same initial conditions when comparing them.

The performance is then evaluated on the voxel representation with commonly used evaluation metrics in anomaly detection.

Chapter 2 will provide some common terminology for anomaly detection and explain some concepts relevant to the image-based anomaly detection method selected. In Chapter 3, the goal is to give an overview of the current state-of-the-art in anomaly detection on the object level. The current trends and potential research gaps for camera and LiDAR anomaly detection will be discussed. Chapter 4 explains the methodology chosen, including how the state-of-the-art methods were selected and the design of the comparison process. Following up, chapter 5 provides details on the retraining of both methods. Eventually, the evaluation results will be discussed in Chapter 6.

2 Background

This chapter will introduce some terminology relevant to the field of anomaly detection. Furthermore, it will provide some explanations for mask-based transformers as I will talk in more detail about these types of models in the context of state-of-the-art research and the methods chosen for this thesis.

2.1 Anomaly Detection in Autonomous Driving

Anomaly Detection is the field of research that focuses on the problem of detecting the unexpected. In one of the pioneering works about anomaly detection[20], the authors define an anomaly as "patterns in data that do not conform to a well-defined notion of normal behavior". This, however, requires a firm definition of the norm that differentiates between normality and every possible anomaly that could appear. In autonomous driving, the application field of this thesis, finding a definition for an anomaly is particularly challenging as there are almost endless possibilities in how a road scenario can derive from the norm. One indicator of this is the large variety of anomaly detection methods for autonomous driving that focus on a specific type of anomaly. Research fields span from detecting lidar spoofing attacks[64] or detecting unknown objects to correctly identifying abnormal or hazardous behavior of traffic participants[47], just to name a few.

2.1.1 Categories in Anomaly Detection

In their work, Breitenstein et al.[13] make an effort to systemize anomalous scenarios in the context of autonomous driving. They distinguish between anomalies on the following levels:

Anomalies on the pixel(/point) level consist of measurements that fall out of the expected range. This could include a pixel in an image with an unexpected value or a point in a lidar scan with an unforeseen distance measurement.

Anomalies on the Domain Level are defined as large shifts in the environment, i.e., the appearance of a snowstorm.

The category of **Anomalies on the Object Level** is comprised of scenarios where unknown instances appear. An unknown instance can be classified as anything that is not categorizable into the semantic classes commonly appearing on the road. Exemplary, any scenario with the sudden appearance of a wild animal or an unknown object, such as a stroller, can be classified into this category.

Anomalies on the Scene Level refer to scenarios where a known object or instance appears in an unexpected place or quantity.

Finally, **Anomalies on the Scenario Level** refer to scenarios where unexpected behavioral patterns appear from an instance. Exemplary here would be a ghost rider scenario.

This thesis will only focus on anomaly detection methods on the object level. In the research literature, this field of study is also often referred to as outlier detection[44, 7], or anomaly segmentation[18].

2.1.2 Open-Set Segmentation and subfields

Most anomaly detection approaches on the object level base their functionality on top of segmentation models. Here, a model needs to segment an image into different regions based on what they represent. In the domain of semantic segmentation, models are usually only given a fixed amount of classes that the image regions can be classified into. This can be problematic when deploying these models in the real world, as they might struggle to handle objects that cannot be classified into one of the known classes. The field of open-set segmentation tries to tackle this problem by training models to recognize unknown objects or instances. In contrast to classical anomaly detection methods, open-set segmentation models will classify regions representing known objects as well. Despite offering better deployment in real-world applications, open-set segmentation models tend to perform weaker for classic segmentation compared to their Closed-Set Segmentation relatives since these methods are not forced to segment a region into a known class.

Some subfields of open-set segmentation include open-set semantic and instance segmentation.

The more recent research field of Open-Set Instance Segmentation[38, 49, 25] requires to detect all instances in the data but not to label them. This differs from anomaly detection, where unknown objects have to be tagged as such.

2.2 Terminology

2.2.1 Mask-based Transformers in Computer Vision

As this thesis explores a mask-based transformer, some concepts that often appear in this context will be introduced.

Transformers are encoder-decoder architectures that utilize an Attention mechanism (figure 2.1). In an attention module, there are three learnable parameters, also referred to as query, key, and value. Similar to a database query, the attention mechanism measures the similarity between the query vector and the key vector. Taking the dot product, we will receive an attention filter matrix to be used on the value vector where unimportant features of the value vector may be filtered out.

$$attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}}) \cdot V$$

where d_k is the dimension of the key.

Mask Classification refers to a paradigm for segmentation problems in computer vision. While Per-Pixel Classification Models predict a probability distribution δ^K over all possible labels K for every single pixel of an image, mask classification models will first split the image into N multiple regions and then predict a probability distribution over all K classes for each region.

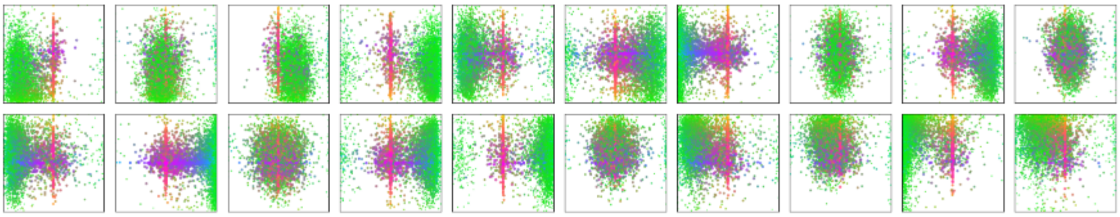


Figure 2.2: object queries tend to specialize in detecting certain features at specific locations of the feature maps. The pictures show different querying behavior from object queries in the DETR model[15]

Most computer vision models for object detection use predefined anchor boxes spread across the input image. In recent years, there has, however, been a surge in transformer models utilizing a different approach:

Object Queries are a set of learnable vectors that often appear in the context of computer vision transformers. The concept has been introduced by Carion et al. for their object detection model called DETR[15]. Each query requests different information from the feature maps. Object queries are, at their core, non-geometric entities, meaning they are not initialized with geometric features they are supposed to detect. Nonetheless, such geometric features can be learned by the object queries over the course of training. Another interesting aspect of object queries that was studied by Carion et al. is the internal interactions between them. Each object query tends to specialize in detecting specific features in specific locations on the feature maps2.2. The process of how these object queries specialize is greatly dependent on the specializations of their counterparts. For example, one object query would favor querying features on the left side of the feature map if one of its counterpart object queries specializes on the right side.

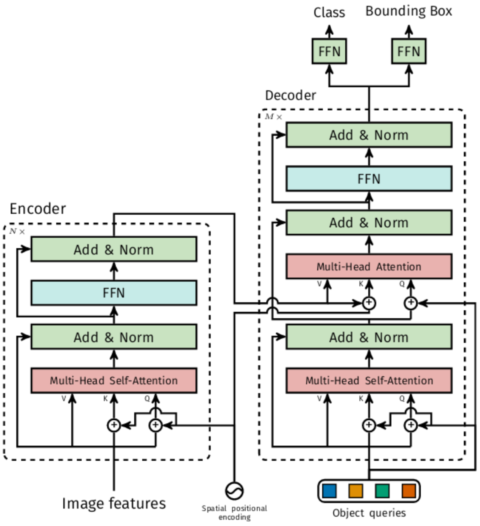
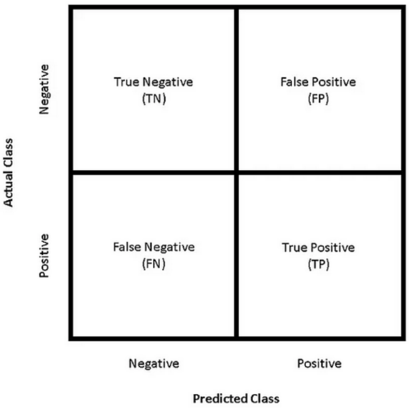


Figure 2.1: Architecture of the transformer module in the DETR model[15]. Image features are processed in the encoder and then fed into the decoder. The N object queries are the main input of the decoder and will output a set of bounding boxes and classes

2.2.2 Metrics

Anomaly Detection on the object level, as explored in this thesis, can abstractly be understood as a binary classification problem. Formally, one could define an anomaly detection method as a function $s(z) : Z \rightarrow [0,1]$ that outputs an anomaly score for each element z (pixel/ point/ voxel) in the dataframe Z (image/ point cloud/ voxel grid). I will define the problem for image-based anomaly detection; the definition can be analogously used for point clouds.



		Predicted Class	
		Negative	Positive
Actual Class	Negative	True Negative (TN)	False Positive (FP)
	Positive	False Negative (FN)	True Positive (TP)

Figure 2.3: Confusion Matrix

We can use binary classification metrics to evaluate the performance of our anomaly detection methods. Such metrics are based on the the values in a confusion matrix2.3 though some of the values are more interesting in the context of anomaly detection.

Strong-performing anomaly detection methods are characterized by their ability to correctly detect the anomaly and, secondly, to correctly differentiate between anomalies and normal instances.

These characteristics can be analyzed particularly well with precision- and recall-based metrics. Precision (also known as PPV, Positive Predictive Value) and Recall (also known as True Positive Rate) are calculated as follows:

$$P = \frac{TP}{TP + FP}$$

$$R = \frac{TP}{TP + FN}$$

These two values can be plotted across a continuum of thresholds in a Precision-Recall Curve.

Calculating the area under this curve will give us a threshold-independent value that tells us a lot about the models' ability to correctly distinguish between anomalies and non-anomalies. The **Area under the Precision-Recall Curve** (AuPRC) is particularly well suited for evaluating a binary classificator's performance on imbalanced data, where the number of positives is considerably lower than the number of negatives. Imbalanced data is almost always to be found in datasets used for evaluating anomaly detection methods. It should be noted that the area under the PRC is equivalent to the average precision metric (AP). The average precision can be interpreted as the precision averaged over all recall values between 0 and 1, which is the same as calculating the integral of the precision-recall curve.

Next to the AuPRC, the **Area under the Receiver-Operating-Characteristic Curve** (AuROC) is the most commonly found metric in anomaly detection. The Receiver Operating Characteristic Curve plots recall against the false positive rate. In the research literature, there is an ongoing discussion about whether AuROC serves as a good metric for describing anomaly detection performance[39, 56] for imbalanced data, as the ROC Curve is less robust to changes in the percentage of negatives appearing in the data than its AuPRC relative. A perfect binary classificator would reach an AuROC of 100%, a binary classificator with an AuROC of 50% is equivalent to a coin flip. The False Positive Rate is calculated as

$$FPR = \frac{FP}{FP + TN}$$

Another metric that can often be found in binary classification evaluation on imbalanced data is the **False Positive Rate at 95% Recall** (FPR$_{95}$) score. It shows percentage-wise how many false positives are needed to reach a total of 95% true positives. As false positives are to be avoided in anomaly detection this metric is also commonly used for evaluating anomaly detection methods. A perfect binary classificator would have a FPR$_{95}$ score of 0%.

Closely related to the AuPRC metric is the **F$_1$ score**, which can be interpreted as the harmonic mean between precision and recall.

$$F_1 = 2 * \frac{P \cdot R}{P + R}$$

Finally, **Specificity** is a measurement that detects the models' ability to correctly identify negative elements, i.e., non-anomalous regions, in the context of anomaly detection. It is calculated by:

$$S = \frac{TN}{TN + FP}$$

2.2.3 Voxel Grids

Voxel Grids are a form of three-dimensional representation that have applications in computer graphics, scene representation[31, 63], and computer vision[65, 54]. In computer graphics, a single voxel is usually defined through its neighboring voxels and its occupancy[29, 41]. For the scope of this thesis, it is, however, advantageous to define the voxel space via three-dimensional coordinates to better illustrate the transformation functionality from point cloud to voxel grid in 4.4.1. The voxel grid V_{grid} can be defined as a bounding box with height H, width W, and depth D. Each voxel v resides in V_{grid} and can be uniquely identified through the coordinates of its center point $v = (x, y, z) \in (D, X, Y)$. Furthermore, every voxel occupies a space of v_{res}^3, where v_{res} is a multiple of X, Y, and D and can be thought of as the resolution of the voxel.

3 State of the Art

The following chapter introduces some current research in the field of anomaly detection in the context of autonomous driving. The goal is to give an overview of the research field and provide some insights into trends and problems the research community favorably works on.

3.1 Anomaly Detection Benchmarks

Advances in the research field of anomaly detection would not happen as effectively if it weren't for benchmarks that anomaly detection methods could be tested on. Benchmarks provide possibilities to evaluate and compare methods to each other in a standardized way whilst indirectly pointing out research gaps that could be filled.

For camera-based methods, the most prominent sensor domain for anomaly detection in autonomous driving, there exist two prominent benchmarks for anomaly detection in the context of autonomous driving at the time of writing this thesis. Segmentmeifyoucan[18] extends the Road Anomaly benchmark introduced in [43]. Their dataset consists of road sceneries with unexpected instances like obstacles or wild animals appearing. Chan et al.[18] argue that whilst precision in object detection for anomalies is important, the key from a practical standpoint is the detection of all anomalous regions in a scene regardless of the size. Pixel-level metrics like the AuPRC or FPR_{95} cannot capture the performance for small anomalies very well. Therefore, the authors propose a set of evaluation metrics that can measure performance better in terms of capturing anomalies as an entire component. These metrics include an adjusted sIoU metric, as well as the Positive Predictive Value and F1 score.

In addition to SegmentMeIfYouCan, most state-of-the-art anomaly detection methods in the camera domain refer to the fishyscapes benchmark[9]. The fishyscapes benchmark extends the lost and found dataset[51] and adjusts frames from the cityscapes validation dataset[23]. The cityscapes validation split is overlayed with anomalous objects whilst the lost and found dataset is cut down to only include images with unknown instances. Similar to SegmentMeIfYouCan, Fishyscapes focuses on precision-based metrics, namely the average precision score. The authors argue that other popular metrics for computer vision, such as AuROC (area under operating receiver characteristic curve), do not match the task of anomaly detection very well because the amount of inlier data and outlier data is unbalanced. Moreover, the number of false positives is more relevant for safety, which is why FPR_{95} is chosen as the secondary evaluation metric. From a practical standpoint, anomaly detection methods should be quick for judgment and not lose performance on regular segmentation tasks. Blum et al.[9] therefore chose to include the mean intersection over union to evaluate segmentation as well as the runtime needed to detect anomalies in a single frame.

In contrast to the camera domain, where a lot of options for evaluation and performance comparison exist, anomaly detection methods in other sensor domains mostly lack common benchmarks that they can be tested on[11]. In the LiDAR domain, the majority of anomaly detection literature additionally introduce a method to generate or label unknown instances and outliers in a closed-world dataset like SemanticKITTI[5] or nuScenes[14].

The Semantic KITTI dataset[5] is based on the KITTI dataset[32] and provides point cloud representations of the sceneries as well as semantic labels. Whilst only including a finite amount of class labels, the authors have introduced an open-set instance segmentation variant of their benchmark[49, 6]. However, as explained in 2.1.2, open-world instance segmentation only measures the performance of class-agnostic instance detection, resulting in methods not distinguishing between known and unknown objects.

Alliegro et al.[3] introduce another benchmark for open-set semantic segmentation, but they only focus on point representations of objects rather than road sceneries.

3.2 Anomaly Detection Methods for Images

Cameras are the easiest and cheapest solution for cars to perceive their surroundings. It is evident that, compared to other sensor domains relevant to autonomous driving, anomaly detection on camera data is the most extensively researched field within anomaly detection. This is most notable when comparing the number of available benchmarks to test image-based OoD-Detection methods in contrast to other sensory data like LiDAR or radar[11].

Anomaly Detection Methods can be broadly categorized into uncertainty-based, reconstructive, and generative methods.

Reconstructive Methods:

Reconstructive methods try to reconstruct the input data and calculate the difference between the synthesized data and the original input. Regions that are resynthesized poorly are then labeled as anomalies. An approach that could be categorized as reconstructive is the model proposed by Di Biase et al.[26]. Their method combines uncertainty estimation with image reconstruction. After the image has been processed through a segmentation network, the uncertainty of a segmented area is calculated via softmax entropy before a synthesis module resynthesizes the segmented image. Finally, a dissimilarity module based on an encoder-decoder model calculates the anomaly predictions. The more dissimilar a segment of the reconstructed image is from the input, the more likely it is that this segment represents an anomaly.

Generative-Based:

The term generative refers to approaches as models that synthesize their own negative/anomalous data for training [35, 34, 11]. For instance, Besnier et al.[7] propose a generative method built on top of a Bayesian Neural Network[33] where adversarial attacks are utilized in training. An observer network observes the results of a semantic segmentation network and is trained to predict the probability that the segmentation networks' output is not aligned with the correct class. The training was designed to fine-tune the model to anomalous instances, which is done through local adversarial attacks where negative data is generated.

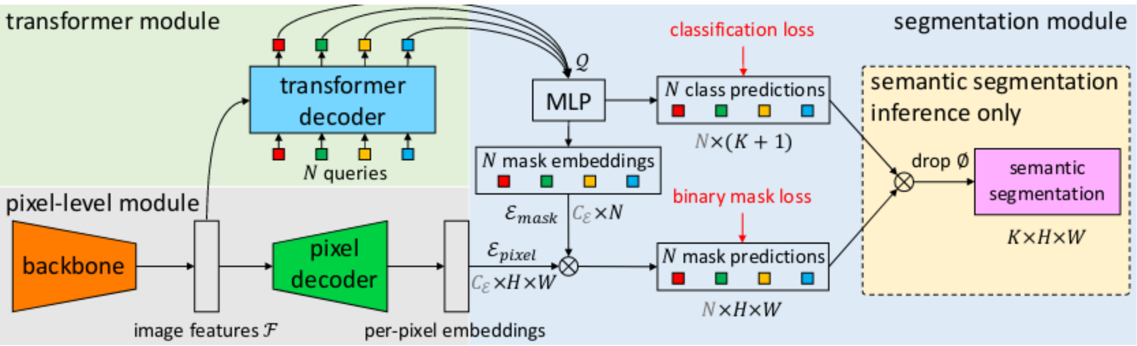


Figure 3.1: Illustration of MaskFormer's[22] architecture. MaskFormer consists of an encoder-decoder module that outputs pixel-wise features and a transformer decoder whose object queries output mask features and class predictions. The class and mask predictions are then combined with a bipartite matching algorithm.

Confidence-Based:

The majority of methods calculate their anomaly prediction scores based on the grade of uncertainty that a closed-world segmentation model has about its predictions.

Galesso et al.[30] use Max Logits combined with a similarity score between the test image and a library of reference features learned by a semantic segmentation network. There, the average distance between each feature in the test image and its k nearest neighbors from the reference features is calculated. The authors also emphasize the potential of transformer features for anomaly detection, especially in combination with their k-nearest-neighbor approach.

Uncertainty can also be modeled with maximized entropy, as Chan et al. show in [19]. Their model is based on a retrained semantic segmentation CNN for higher entropy scores when detecting unknown objects. The loss function differentiates between inliers and outliers, where a cross-entropy loss function is applied for in-distribution samples while a negative log-likelihood over the average of all classes is used for outliers. Minimizing the loss function for outliers, however, is equivalent to maximizing the softmax entropy. A segment is predicted as out-of-distribution if it exceeds a certain entropy threshold. Additionally, a meta-classifier will help remove false predictions by evaluating aggregated dispersion measures and geometric features of the predicted segments.

3.2.1 Mask-based Anomaly Detection

In recent months, anomaly detection methods built on top of mask classifiers have emerged as a strong-performing approach, setting a new state of the art in the image domain. Evaluation rankings for prominent anomaly segmentation benchmarks such as SegmentMeIfYouCan[18] and the fishyscapes benchmark[9] are, at the time of August 2023, clearly dominated by mask-classifiers.

All of the following methods exploit the object query mechanism for mask predictions in the Mask2Former[21] model. This being the case, it is important to first understand how Mask2Former is able to accurately predict masks for semantic segmentation.

Mask2Former and its predecessor MaskFormer[22] consist of two modules, namely a pixel encoder-decoder for extracting per-pixel embeddings and a transformer decoder that outputs mask

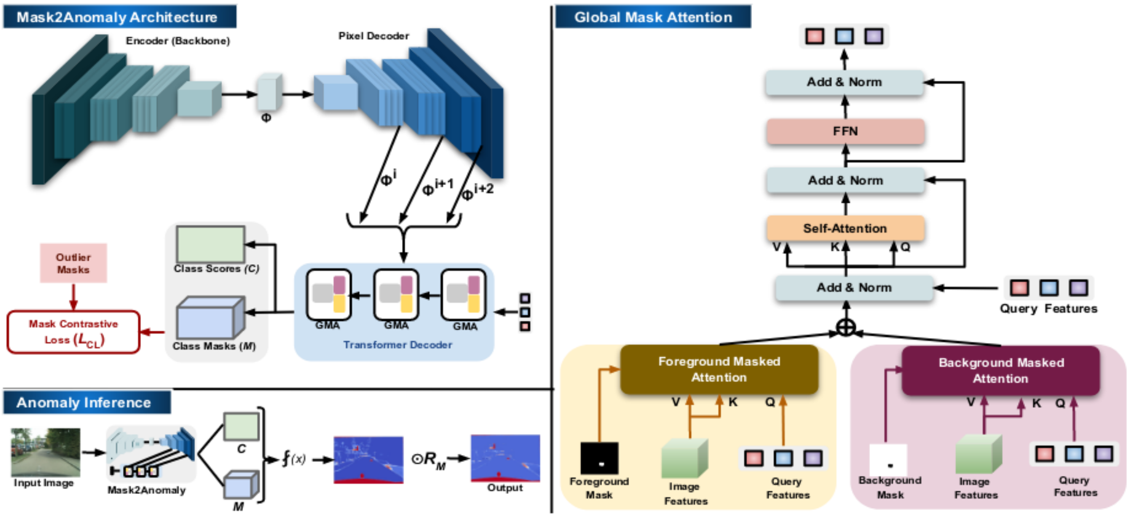


Figure 3.2: Mask2Anomaly architecture[55]

embeddings and their class predictions. The pixel decoders' embeddings are fed at the input layer into the transformer decoder and are additionally used to formulate mask predictions on the image. The transformer decoder takes so-called "object queries" as inputs in a parallel manner. These learnable feature vectors were first introduced in DETR[15]. The queries will predict a set of binary masks next to a set of mask assignment predictions that match a class to each mask found. Every query tends to specialize in recognizing one specific class across a certain region in the input image. Mask2Former's loss function is calculated through bipartite matching between the predicted masks and the ground truth masks. Given a set of predictions $z = \{(p_i, m_i)\}_{i=1}^{N}$, with p being a probability distribution over K classes and the void class, and a set of ground truth segments $z^{gt} = \{(c_i^{gt}, m_i^{gt}) | c_i^{gt} \in \{1, ..., K\}, m_i^{gt} \in \{0,1\}^{H\times W}\}$ the mask classification loss function is a combination of a binary mask loss function $\mathscr{L}_{mask}$ and a cross-entropy loss function: $\mathscr{L}_{\text{mask-cls}}\left(z, z^{\text{gt}}\right) = \sum_{j=1}^{N}\left[-\log p_{\sigma(j)}\left(c_j^{\text{gt}}\right) + \mathbb{1}_{c_j^{\text{gt}} \neq \varnothing} \mathscr{L}_{\text{mask}}\left(m_{\sigma(j)}, m_j^{\text{gt}}\right)\right]$. The loss value depends on how well the mask probability distribution matches the ground truth class of the matched segment. For inference, MaskFormer assigns each pixel to one of the predicted probability masks that it is covered by.

Mask2Former improves on top of MaskFormer in speed by feeding the pixel-decoder features to one transformer decoder layer at a time, as well as extending its application to instance segmentation and panoptic segmentation. Notably, Mask2Former is extended in functionality, for instance and panoptic segmentation.

Predicting masks for segmentation, as Mask2Former does, has a number of distinct properties that can help detect anomalies more accurately. In contrast to pixel-wise anomaly detection methods, where each pixel is given an anomaly score individually, masking helps detecting an anomalous object as a whole entity rather than a cluster of individual pixels with their own individual anomaly score. Specifically for Mask2Former, the model on which most of the state-of-the-art mask anomaly detection methods are based, all masks found are classified into K known classes, whilst the assignment predictions for pixels to masks are not mutually exclusive. Grcic et al., authors of a Mask2Former-based Anomaly Detection method[36], state that this can be used to their advantage as their model can more easily reject entire masks and, therefore, reject entire instances rather than discriminating clusters of singular pixels. They predict that the masking paradigm may lead to a reduction of false positives since each pixel in the anomaly mask is treated equally.

Ackermann et al., another team of authors that utilize Mask2Former for Anomaly Detection[1], capitalize on a specific property of Mask2Former for their approach. They argue that mask-based models trained for semantic segmentation already learn to assign certain masks to unknown objects. This trait can effectively be exploited when fine-tuning a model for anomaly detection. Ackermann et al. speculate that these specialized masks for unknown objects can be found by finding the query in the attention mechanism of Mask2Former, which specializes in predicting unknown objects. This query was then extracted by comparing the average IoU of each mask with the anomalies in a validation dataset and retracing the object queries responsible for predicting the most overlapping masks. Thus, the anomaly predictions of the model are equal to the mask prediction that the specialized query predicts.

Similarly, Nayal et al.[48] argue that the transformer architecture works well for anomaly segmentation as the "object queries [which are the specialized inputs for the transformer decoder] tend to behave like one vs. all classifiers." Nayal et al.'s method will be further elaborated in the methodology chapter (4), as it was chosen for my thesis.

Grcic et al.[36] use a slightly different approach they call "ensemble over anomaly masks" (EAM). Their model calculates anomaly scores based on all mask predictions. For every pixel, it considers all results of all masks covering this area. Every class assignment prediction for these relevant masks is then aggregated to formulate an uncertainty score.

The most recent method that builds on top of Mask2Former is Rai et al.'s "Mask2Anomaly" method[55]. Mask2Anomaly provides a novel contrastive learning approach where the model is motivated to differentiate between predicting high anomaly scores for normal and anomalous regions. The training included the use of synthetically generated OOD data, where snippets of anomaly regions from anomaly segmentation datasets were pasted into images. The anomaly score for each pixel is computed through $f(x) = 1 - max(S(x))$, with $S(x)$ representing the pixel-wise class scores.

3.3 Anomaly Detection for LiDAR

Anomaly Detection for LiDAR data sets itself apart in the central themes the literature tends to explore compared to research in image-based anomaly detection. Whilst there is a lot of research on anomaly segmentation for images to be found, research on LiDAR data extends into anomaly detection on the point level (2.1.1), like the identification of adversarial attacks[37, 64, 2] or the neutralization of distortions in the point cloud caused by bad weather[52]. Research into anomaly detection in the LiDAR domain appears to adapt to the sensor's weaknesses and advantages. This would also explain the focus on developing class-agnostic methods that frequently appear in the LiDAR domain. As LiDAR scans do not provide the level of fine-grained detail a camera can offer, semantically labeling regions in a point cloud will be more challenging than classifying segments of an image. They can, however, outline instances better in a three-dimensional space than a camera due to their accurate depth measurements.

Methods that focus on the detection of anomalous objects in LiDAR scans will, for convention purposes, be classified into class-agnostic and class-aware.

3.3.1 Class-Agnostic

These methods do not classify instances according to the correct class label but instead assign instance-wise labels, ignoring the region's class semantics. All state-of-the-art point cloud methods specializing in open-set instance segmentation are, to my knowledge, class-agnostic.

Deng et al.[25] propose a class-agnostic method for open instance segmentation that clusters neighbors in an ellipsoidal shape. They argue that this better matches the nature of LiDAR scans since "the farther a point is from the LiDAR sensor, the larger the distance from its neighbors is in vertical and horizontal directions" (Section 3.B). Foreground points are separated from background points by a panoptic segmentation network. The unknown points are then clustered to unknown instances through an ellipsoidal neighbor search. ElC-OIS currently ranks as the strongest performing method on the SemanticKITTI Open Set Instance Segmentation split[6] with an IoU of 0.691 for unknown objects.

Nunes et al.[49] build on top of their semantic segmentation network SegContrast[50]. The point cloud is first augmented by clustering into background and foreground points via ground segmentation. The non-ground points are then clustered via HDBSCAN to formulate instance proposals. These proposals, however, are not refined and need further processing. The point cloud is first processed through a series of clustering methods, including a ground segmentation module that divides between foreground and background points. HDBSCAN helps formulate instance proposals in the foreground region whilst regions of interest and point features are extracted through the pretrained semantic segmentation network and saliency maps[57]. The actual instance segmentation is then computed with a GraphCut implementation. Each point is represented as a node whilst weighted edges are computed between all neighbors. The GraphCut algorithm will terminate all weak edges, resulting in connections only between neighbors that most likely belong to the same instance.

3.3.2 Class-Aware

Reconstructive Methods:

A more recent paper by Piroli et al.[53] explores outlier synthesis in latent space. To correctly insert outliers, the density of the inlier distribution in the features first has to be estimated by an auto-encoder. The core of this method is a reconstruction step in the pipeline where a noise vector is added to the encoded features. These noisy features will then be reconstructed to produce virtual outliers. Thus, an uncertainty estimation head is trained with a binary sigmoid loss function using these synthesized outliers to correctly identify out-of-distribution data.

Confidence-Based:

To my knowledge, there is currently only one open-sourced method that provides labeling of unknown instances for 3D LiDAR scans. Cen et al.[17] propose a framework that can do outlier-aware semantic segmentation and incremental learning of novel classes. Their method is built on top of the Cylinder3D[66] framework that is specialized on closed-set semantic segmentation. Details about this method will be discussed in further detail in 4.3 as it was chosen for evaluation.

Similarly, Najibi et al.[47] point out the research gap in open-set detection for LiDAR scans though their model also focuses on trajectory prediction of unknown instances rather than detection. Their method consists of two unsupervised modules, namely, a scene flow estimator and an auto meta-labeling module. The unsupervised scene flow estimator approximates local scene flow through scene decomposition, where the scene is separated into connected components. The auto meta-labeling module then reconstructs the shapes of these components and predicts labels for them. According to the authors, their method outperforms other unsupervised approaches for flow estimation and can hold its ground against supervised approaches.

Li and Dong[42] propose an open-world semantic segmentation network that utilizes a feature extractor in combination with a generative adversarial network for semantic understanding. Their method, named APF, learns discriminative prototypes for every class through clustering approaches. Thus, segmentation predictions are based on a distance measurement between point features and the prototype features. To detect unknown classes, APF is dependent on its GAN module, where the adversarial mapper estimates the characteristics of unseen classes. For inference, the model considers features from the feature extractor as well as the class predictions from the adversarial mapper in the GAN module. Li and Dong base their outlier extraction on the assumption that outliers aggregate in the center of the learned feature space. Therefore their sum over distances to all learned prototypes must be smaller than for known classes.

3.4 Findings

Overall, in the field of camera-based anomaly detection for autonomous driving, it is likely that more transformer-based will appear in the near future. Namely, the masking mechanism of Mask2Former[21] has been utilized multiple times for anomaly detection to great success, but there are other strong-performing transformer-based approaches, such as [30]. Reconstructive approaches, though still very relevant in the field, are currently dominated by uncertainty-based anomaly detection methods when comparing performance on the benchmarks SegmentMeIfYouCan[18] and Fishyscapes[9]. SynBoost[26], the reconstructive method with the highest AuPRC score on fishyscapes and SegmentMeIfYouCan, does not rank in the top ten for both benchmarks. One current discussion point in the field of anomaly detection is the usage of outlier data to further sensitize the model to anomalous instances. Extending this further, most methods classified as generative base their entire training strategy on feeding the model with self-synthesized outliers. It should, however, be noted that the majority of the literature reports an increase in performance after fine-tuning their anomaly detection model[19, 36, 44].

Anomaly detection on the object level for LiDAR data has not been explored as thoroughly as in the camera domain. Whilst image recognition has historically been the primary research focus in the field of computer vision, this also is likely to be attributed to a lack of standardized benchmarks on which anomaly detection methods in the LiDAR domain can be tested. In contrast to the camera domain, anomaly detection for LiDAR scans is often set in relation to anomalies on the point level, such as adversarial attacks or weather-based effects.

	Year	Hardware	Corner Case	Fine-tuned	Approach
3DUIS[49]	2022	LiDAR	Open Instance Segmentation	No	Confidence-based
APF[42]	2023	LiDAR	Open Semantic Segmentation	No	Confidence-based Generative
cDNP[30]	2023	Camera	Object Level AD	No	Confidence-based
EAM[36]	2023	Camera	Object Level AD	Yes	Confidence-based
ElC-OIS [25]	2023	LiDAR	Open Instance Segmentation	No	Confidence-based
LS-VOS[53]	2023	LiDAR	Object Level AD	No	Reconstructive
Mask2Anomaly[55]	2023	Camera	Object Level AD	Yes	Generative
Maskomaly[1]	2023	Camera	Object Level AD	No	Confidence-based
Maximized Entropy[19]	2021	Camera	Object Level AD	Yes	Confidence-based
Najibi et al.[47]	2022	LiDAR	Object Level AD	No	Confidence-based
ObsNet[7]	2021	Camera	Object Level AD	No	Generative
RbA[48]	2023	Camera	Object Level AD	Both	Confidence-based
ReaL[17]	2022	LiDAR	Open Semantic Segmentation/ Novel-Class Discovery	Both	Confidence-based Generative
SynBoost[26]	2021	Camera	Object Level AD	Yes	Reconstructive

Table 3.1: A comparison of all state-of-the-art methods that were introduced in this chapter. The column "Fine-tuned" refers to whether the method was trained with outliers. Details on how the methods were selected can be found in 4.1.1

4 Method

This chapter will give an overview of the necessary steps to allow a fair benchmarking process of the chosen methods. I will first explain the thought process behind choosing the models that were benchmarked in this thesis, as well as the models themselves. The two models were chosen from the current state-of-the-art of anomaly detection across the camera and LiDAR sensor domain. To allow equal initialization conditions for the evaluation, both methods were retrained on the same dataset. Finally, the evaluation pipeline and each of its components, namely the dataset design and the voxelization method, will be detailed out.

4.1 Model Selection

This section will explain the approach taken in the search for methods to be benchmarked for this thesis. It should be noted that all state-of-the-art methods presented in the previous chapter 3 were acquired simultaneously while searching for the two methods to be selected for the evaluation. Thus, the selection of state-of-the-art methods was obtained through the same search methods described in 4.1.1.

4.1.1 Search Criteria

Methods were initially searched with the keywords "Anomaly Detection", "Outlier Detection", and "Unknown Object" in combination with "Autonomous Driving". For finding state-of-the-art anomaly detection, citation search is more effective than the snowball search method. The majority of camera-based methods found through these search strategies evaluated their methods' performance either on the SegmentMeIfYouCan[18] or Fishyscapes[9] benchmarks. As benchmarks are particularly useful for directly comparing the performance of methods, further research in the camera domain was narrowed down to submissions that appeared in their leaderboards.

For research on LiDAR anomaly detection methods, the same keywords from the search in the camera domain were used in addition to "LiDAR" and "Point Cloud". The term "Anomaly", however, is frequently associated with fragmented point cloud representations[46, 28]. Therefore, I expanded the set of search keywords by "Open-Set Segmentation" and considered methods introduced in survey papers, such as in [11, 59]. Although class-agnostic methods for open-set instance segmentation are not relevant to this thesis, they are often cited in papers presenting potential candidates to benchmark. Exemplary, Najibi et al.'s method[47] was found through a citation search on a well-known paper on class-agnostic open-set instance segmentation by Wong et al.[61].

The Anomaly Detection methods searched for had to fulfill the following requirements to an extent:

Accessibility:

Open-source methods with good documentation and support for reproducing the evaluation results presented in the paper were preferred. Some good indicators of a likely successful implementation include a detailed description of the experiment procedures, whether checkpoints are provided, and whether the model provides support for training on a custom dataset. The generated training data set prepared for this thesis uses cityscapes[23] labels in its ground truth. Any model trained on the cityscapes benchmark is likely more adjustable for custom training, as the corresponding cityscapes data loader implemented in the code base can be used. A successful implementation is also more probable if the results of the model have been correctly reproduced as part of the evaluation in another paper. Models built with PyTorch are preferred.

Performance:

The methods were compared on performance claims in the paper and rankings in leaderboards for anomaly detection benchmarks.

For image-based models, methods evaluated on the SegmentMeIfYouCan[18] and Fishyscapes[9] benchmarks were mainly focused on. In their survey, Bogdoll et al.[11] point to NFlowJS[34] as the best-performing method for image-based anomaly detection across all benchmarks. As NFlowJS was released 2 years prior to the time of writing this thesis, its results on the main metrics AP and $\mathrm{FPR}_9 5$ for the benchmarks SegmentMeIfYouCan and Fishyscapes were chosen as a guideline value. Another good point of reference for strong performance is the influence a particular method has on the field of anomaly detection in autonomous driving. While this cannot be accurately measured, some good indicators for the influence a paper has, are the number of citations or the prestige of the conference the paper was accepted into. I particularly looked into papers accepted to the Computer Vision and Pattern Recognition Conference (CVPR), the European Conference on Computer Vision (ECCV), the International Conference on Computer Vision (ICCV), the International Conference on Pattern Recognition (ICPR), and the IEEE Winter Conference on Applications of Computer Vision (WACV).

As previously mentioned, anomaly detection for LiDAR methods has, to my knowledge, no standardized benchmark as of the time of writing this thesis. Hence, the research mainly relied on performance claims the authors made in their papers. This was unfavorable, as most authors used a private anomaly benchmark to test their method, and performances could not be compared accurately. The authors of the SemanticKITTI[5] benchmark, one of the most popular 3D datasets for road sceneries, released a variant of their dataset with unknown object instances[6]. However, all methods that were submitted to this competition at the time of writing this thesis specialized in class-agnostic instance segmentation. Consequently, methods submitted for this competition could not be used for evaluation on this thesis' benchmark.

Age and Relevancy: Due to the increase in overall performance of anomaly detection methods that came along with the introduction of vision transformers in 2020, previously published methods may be considered dated and thus no longer considered state-of-the-art. Therefore, methods like ImageResynthesis[43](released in 2019), although displaying solid results on SegmentMeIfYouCan and having a great influence on the field of anomaly detection in autonomous driving by introducing a paradigm shift to reconstructive approaches, were excluded from my search.

Open-sourced LiDAR methods for anomaly detection on the object level are less common than open-sourced methods for images. Therefore, I refrained from excluding anomaly detection methods on point clouds due to their age.

Exclusion of Methods fine-tuned on anomalies in training: For this research, methods that were fine-tuned on data that shows anomalous instances or objects were excluded. Unlike a problem or game within a closed setting like a crossword puzzle or chess, the problem of autonomous driving is open-ended since there is an endless amount of possible scenarios that an autonomous vehicle could encounter on the road. Similarly, there are an uncountable number of unknown objects or instances that an autonomous vehicle may come into contact with. For that reason it is impossible to train a vision classification system such that it will be prepared for every situation that it encounters. When fine-tuning an anomaly detection method with anomalous instances the model is trained to learn a closed set of anomalies that can possibly appear in a road scene. This, however, is counterintuitive to the open-set problem definition. Consequently, I refrain from benchmarking methods that were fine-tuned on anomalous data.

The majority of anomaly detection methods for autonomous driving utilize fine-tuning. Models with promising results, such as EAM[36] or Maximized Entropy[19] are therefore discarded. Although the authors of RbA also utilize fine-tuning, they additionally provide results for an unsupervised model that achieves strong performance on the SegmentMeIfYouCan benchmark as well.

4.1.2 Selection

From the methods that were named in 3.2 and satisfy the conditions stated above, namely ObsNet[7] and RbA[48], RbA beats out the other contestant performance-wise on the SegmentMeIfYouCan benchmark[18]. Additionally, RbA is built on the well-documented Detectron2[62] framework. Galesso et al.'s cDNP[30] would have been another candidate, but its' codebase was only released in the later stages of this thesis.

Similarly, for the LiDAR method, I focused on methods with great accessibility and strong performance. As previously mentioned, anomaly detection for LiDAR methods has, to my knowledge, no standardized benchmark as of the time of writing this thesis. The chosen method should preferably be tested on road scenarios and utilize no fine-tuning on anomalous data. The anomalies this thesis focuses on are unknown objects that do not usually appear on the road. Accordingly, LiDAR anomaly detection methods that focus on adversarial attack detection[64] or are specialized in Industrial Anomaly Detection[60, 28, 46] will not be used. Whilst being able to detect anomalies on road sceneries, class-agnostic methods like Nunes et al.'s 3DUIS[49] or Deng et al.'s ElC-OIS[25] models cannot differentiate between anomalous and non-anomalous instances. Since this category of methods is not able to classify the instances it detects into inliers and outliers, they were also discarded in the selection process.

The only method that has a publicly available codebase and was tested specifically for the detection of anomalous objects on road sceneries is Cen et al.'s model[17], which specializes in Open-World Semantic Segmentation for LiDAR Scans. ReaL was trained with multiple approaches. One of these utilizes synthetically generated anomalies. For the other approach, the loss function of the

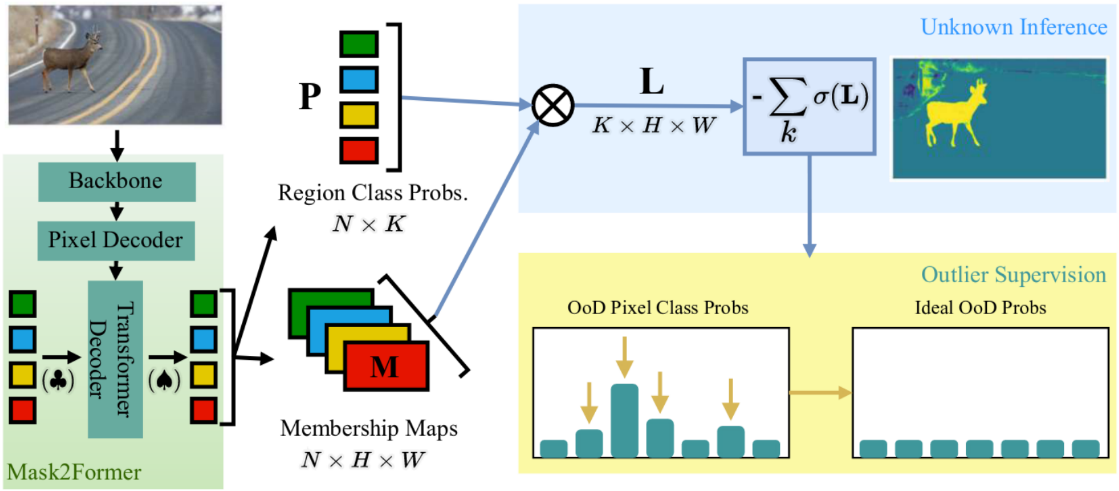


Figure 4.1: Pipeline of Rejected by All[48]. Mask predictions, as well as the class probabilities, are extracted from the Mask2Former's[21] object queries. The outlier scoring function aggregates class logits for all masks.

model is recalibrated to detect unknown objects with a higher probability whilst being trained on normal data. Whilst the former uses negative data and therefore classifies as fine-tuning, the latter approach only uses normal data for training.

4.2 Rejected by All

The camera method is built on top of the Mask2Former model and utilizes the logit scores and region predictions of its object queries. Recall that the Mask2Former architecture consists of an encoder-decoder model that extracts from an image $x \in \mathbb{R}^{H \times W \times 3}$ a set of pixel-wise features $F(x) \in \mathbb{R}^{C_\varepsilon \times H \times W}$, as well as a transformer decoder where Object Queries $Q \in \mathbb{R}^{N \times C_q}$ are learned. Each Object Query will be processed in a 3-layer Multi-Layer-Perceptron (MLP) to predict N binary masks $M(x) \in \mathbb{R}^{N \times H \times W}$ together with the per-pixel-wise features that were extracted from the pixel decoder. Furthermore, the MLP-processed Object Queries predict class assignment probabilities $P_k(x) \in [0,1]^N$ for the masks with k representing one of the K class labels. An illustrative overview of RbA can be found in figure 4.1.

Nayal et al., the authors of RbA, presume that object queries tend to specialize in predicting one type of class each. They propose that outliers can be found in the latent space where the input is rejected by all object queries specialized on known classes. The outlier scoring function is defined as an aggregation over all per class logits of the K queries. Outliers are rejected by all other known classes. The scoring function can be defined as $RbA(x) = 1 - \Sigma_{k=1}^{K} p(y=k|x)$ which can be rewritten without the 1 as $RbA(x) = -\Sigma_{k=1}^{K} \sigma(L_k(x))$ where $L_k(x)$ represents the logits or class scores of class k.

Nayal et al. show, as one of the first, that the mask classification paradigm works well in combination with the detection of unknown objects. RbA exceeds in recall-based metrics like AuROC, as can be observed from the results on the RoadAnomaly and Fishyscapes Lost&Found

benchmark Nayal et al. provide. This can most likely be attributed to the masking mechanism of Mask2Former. The outlier-fine-tuned RbA model currently (November 2023) holds the top spot on the SegmentMeIfYouCan benchmark[18] for the main metric AuPRC with a score of 94.46% and a FPR_{95} score of 4.60%. The RbA model trained with non-anomalous data achieves an AuPRC score of 86.13% and a FPR_{95} score of 15.94%. It currently ranks 6th on the SegmentMeIfYouCan benchmark for AuPRC performance. Whilst keeping strong recall performance, RbA's model trained on non-anomalous data falls back in precision-based scores compared to other state-of-the-art methods with a mean f1 score of 42.04% and a positive predictive value of 41.35% [48].

4.3 ReaL

The LiDAR method chosen for the evaluation is introduced in a paper by Cen et al.[17]. Cen et al.[17] propose an approach that can do outlier-aware semantic segmentation and incremental learning of novel classes. Their method is built on top of the Cylinder3D[66] framework that is specialized on closed-set semantic segmentation. The raw LiDAR scan is fed into a feature extractor before a classifier outputs labels for each point. To sensibilize the Cylinder3D architecture for unknown instances, Cen et al. propose two different approaches for training. The first training method utilizes a mechanism to synthesize unknown instances to approximate the distribution of actual novel objects. This is done by resizing already existing objects. The authors argue that keeping the geometric shape makes it easier to identify the instance as an object rather than noise in the LiDAR scan while still being considered out-of-distribution. For the second training method, which the authors named Predictive Distribution Calibration, no synthetically generated outliers are needed. The classifier is adjusted such that the predictions for points representing known classes are highest for the correct class and second highest for the unknown class. The loss function is defined as:

$$L^{OSeg} = (l(M(\mathbf{P}_{nm}), \mathbf{Y}_{nm}) + \lambda_{cal} l(M(\mathbf{P}_{nm})\mathbf{Y}_{nm}, 0)) + (\lambda_{syn} l(M(\mathbf{P}_{syn}), 0))$$

ReaL bases its outlier scoring function on the uncertainty degree extracted from the class logits. Consequently, the inference function is defined as follows:

$$\hat{\mathbf{Y}}_{\text{open}} = \begin{cases} \underset{i=1,2,\ldots,C}{\arg\max} & g_{nm}(f(\mathbf{P}))\lambda_{,conf} < \lambda_{\text{th}} \\ 0 & \text{otherwise} \end{cases}$$

A point in the point cloud will be labeled by the class with the highest prediction value in the class distribution if it exceeds a certain threshold λ_{th}. Otherwise, the point will be labeled as an outlier. ReaL's performance is lower compared to its counterparts in the camera domain. The Open-Set Segmentation Model achieves an average precision of 20.8% on a SemanticKITTI validation split extended with unknown instances. The method, however, does not significantly drop in detection performance for the closed-world segmentation problem, as can usually be observed for open-world-aware segmentation models.

4.4 Voxel-Based Environment Representation

The goal of this thesis is to compare the performance of methods from different sensory domains. This requires a common ground on which the predictions from both camera and LiDAR domain can be transferred into. Voxel Grids were selected as the common representation to be evaluated on. The modeling of the voxel grid structure chosen for the evaluation process can be seen in 2.2.3.

Voxel representations have certain characteristics that make them ideal for a common representation of the car's surroundings. The 3D structure allows modeling valuable information on distances across its space, unlike image representations. In comparison to point representations, voxel grids are less prone to computation overheads due to ordered storage and can be more storage efficient when maintained in an octree or sparse voxel grid structure[31]. It should, however, be mentioned that voxel grids are more prone to losing fine-grained information about objects compared to both image and point representations.

The voxel transformation method is based on a method that has been developed in the context of a different project at the FZI Research Center for Information Technology. The method was been deployed in the AnoVox benchmark[10] for the transformation of the ground truth information into voxels and, thus, will also be used for the evaluation pipeline of this thesis.

In order to voxelize the image results of RbA, they first have to be transformed into a point cloud.

Given a pixel $p = (u, v) \in \mathbb{N}^{H \times W}$, this can be achieved by multiplying

$$\text{every pixel and their distance } \begin{pmatrix} u \\ v \\ d \end{pmatrix} \text{ with the calibration matrix } \begin{pmatrix} f_u & 0 & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{pmatrix}$$

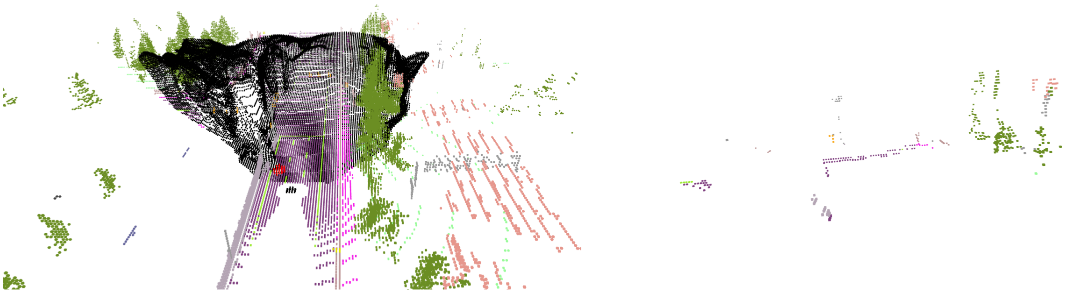
where f_i represent the focal lengths and c_i represent the center point of the camera.

4.4.1 Voxelization of Point Clouds

Given a point cloud P, the 3D space is split into a set of voxels V within a bounding box of height H, width W, and depth D, along the axis of X, Y and Z. Accordingly, each voxel must have a resolution that is a divisible of the bounding box range. Each voxel $v \in V$ contains a subset of points P_v from the point cloud. The value $f(v)$ that the voxel will inherit is dependent on the values of the points $f(p_v)$ in that voxel. The determination of that value differs between the ground truth voxel transformation and the prediction voxel transformation.

Evaluation is done on a voxel grid with a bounding box size of $(1000m, 1000m, 64m)$. Each voxel has a resolution of $0.5m$.

For the ground truth voxel transformation, each voxel inherits the ground truth label of the point closest to the center of that voxel, so $f(v) = \{f(p'_v) | p'_v = min\{||v - p_v|| | p_v \in P_v\}\}$ While this assures accurate labeling for ground truth, this approach has some drawbacks when applied to the anomaly score predictions. Given a voxel v in the prediction with a set of points $P \subset v_{cube}$ located in v where all points except p_{center} were to have a high anomaly score, the anomaly score assigned



(a) Union grid of depth estimator prediction and ground truth

(b) The intersected voxel grid between ground truth and prediction from the depth estimator module

Figure 4.2: A Checkpoint from ZoeDepth[8] pretrained on KITTI[32] was chosen to estimate absolute depth for the voxel transformation. The voxels generated from ZoeDepth are colored in black, the anomaly is labeled in red. As can be seen, the intersection between ZoeDepth's estimated voxels and the ground truth voxel grid averaged a voxel amount less than 10% of the voxel grid intersected with ground truth depth. Moreover, no voxels between the anomalous object in the depth prediction and the ground truth are intersected, as observable in the intersected grid.

to the voxel would wholly ignore the other predictions of $P \setminus p_{center}$.

Therefore, evaluation for transformed voxel grids was additionally provided, where the anomaly score is calculated as the mean of all point scores included, so $f(v) = \overline{f(P_v)}$. This ensures that all predictions the model has made in the image domain are considered when transforming into the 3D domain. A visualization of the entire voxelization process can be found in 4.3.

To minimize computational resource demands for the data generation process, voxels created from the image are cut off if they exceed a distance of over 100 meters. This holds true for both generating the ground truth voxel grid and the anomaly score voxel grid.

4.4.2 Depth Estimator Module

From a practical perspective, it would be interesting to test the camera results on the voxel domain with a depth estimator to estimate distance d for every pixel.

When transferring images from a camera into a 3D representation, an autonomous vehicle would be reliant on a depth estimator. Depth estimator modules can be categorized into absolute and relative depth estimators. Absolute depth refers to a fixed scale (in meters) that the module provides for every pixel, while a relative depth estimator will base its depth predictions on a relative distance scale. For the thesis, the absolute depth estimator ZoeDepth[8] was selected. The authors include a checkpoint that was trained on KITTI[32], a dataset with images of road scenes.

The resulting voxels extracted from the depth estimator's predictions considerably differed in scale from the ground truth depth. Exemplary, the number of voxels taken from intersecting the ground truth voxel grid and the depth estimator's voxel grid in 4.2 was 429, which is considerably lower than the total of 16010 voxels extracted from the ground truth. The scale of distortion made the evaluation with the depth estimator predictions obsolete. A possible solution to adapt ZoeDepth to the CARLA environment would have been retraining on training data generated with the CARLA simulator . This was, however, not possible due to the time limit of this thesis.

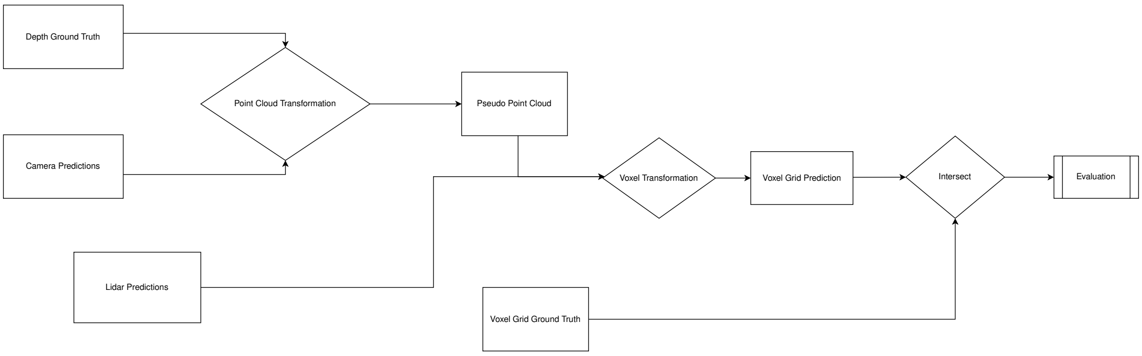


Figure 4.3: Diagram of the evaluation pipeline structure. Anomaly score predictions are evaluated for camera- and LiDAR-based methods. Image scores are transformed into a pseudo point cloud with depth values provided by the evaluation benchmark. The point cloud will be voxelized and intersected with the ground truth voxel grid for evaluation.

4.5 Dataset Modeling

This section will provide information on the generation process of the evaluation dataset and showcase the labeling process for the training dataset.

4.5.1 AnoVox

Bogdoll et al.[12] point out that anomaly detection for other sensory domains, such as LiDAR, falls behind the camera anomaly detection domain due to a lack of OOD-detection benchmarks. In the context of another project at the FZI Research Center of Information Technology, we developed the AnoVox benchmark[10] that tries to bridge this gap by providing both 2D and 3D data for the same road scenarios. This may allow a fair comparison of anomaly detection methods for different sensor domains.

We developed a dataset generator built with the CARLA Simulator[27] that focuses on the generation of abnormal road scenarios. The benchmark we provide consists of ten scenarios as a collection of 185 frames, equivalent to a sequence length of 18.5 seconds, where images and LiDAR scans are collected in each frame. AnoVox provides ground truth labels for all data collected next to depth ground truth for images and a ground truth voxel representation of each frame. In each scenario, the ego vehicle drives around one of ten CARLA towns. In the course of the scene, the vehicle will encounter an anomalous object on the street that was randomly chosen from a collection of objects that were custom-implemented.

Thus, the spawnable anomalies in AnoVox cannot be found in any other CARLA environments, so they cannot be known to an agent training on a CARLA-based dataset with inliers only. The anomalous instances vary in a wide range of different sizes and contexts. Accordingly, the agent may face anything from a small animal to a hot-air balloon in the scenario. Furthermore, every scenario stores detailed information on the spawned anomalous object, such as speed and acceleration in every frame, as well as the size of the anomaly. This allows to group prediction results in the evaluation based on the size of the anomaly that is supposed to be detected.

4.5.2 Generating Training Data

For the scope of this thesis the AnoVox benchmark has been adjusted to allow data generation of scenarios where no anomalies are spawned. Thus, the training datasets used in this thesis were generated with the AnoVox codebase.

All datasets that were used to train RbA[48], were converted into cityscapes' train id label format[24]. RbA's training is limited to 19 classes which did not include some classes that appeared in the evaluation dataset of this thesis. To match the model configuration for the training that the authors propose in their respective paper, the left-out classes ("bridge", "ground", "guard rail") were mapped to classes that RbA acknowledges in training ("wall", "sidewalk", and "fence"). Similarly, ReaL uses a mapping function to group all possible labels into 19 distinct classes. Both remapping functions can be found in the codebase.

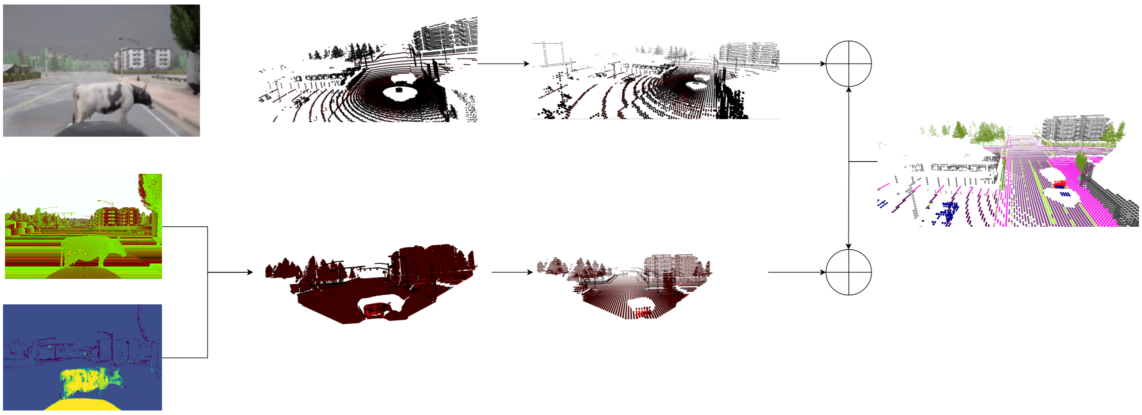


Figure 4.4: Illustration of the Evaluation Pipeline in the following order: Image with Anomaly Scores and Depth Image, Pseudo Point Cloud, Voxel Grid with Anomaly Scores, Intersection and Evaluation with the Voxel Grid containing ground truth labels.

5 Experiment Setup

The experiment chapter will describe the training process in detail. The design of the training set and the selection of the training parameters will be explained.

This thesis pursues to compare and evaluate current state-of-the-art models for anomaly detection in camera and LiDAR detection. The goal is to compare methods as fairly as possible. This requires training and evaluation on the same datasets with the same definition of normality.

5.1 Training Data

The chosen models are trained and evaluated on data generated with the CARLA Simulator[27]. I readjusted the code for AnoVox to generate scenarios with a closed-world environment setting, meaning no anomalies or unknown objects appear in this dataset. Though many state-of-the-art anomaly detection models utilize fine-tuning with outliers to improve performance, one may argue that anomaly detection should not rely on out-of-distribution data for training.

Three datasets $\mathscr{D}^{19130}_{train-static}$, $\mathscr{D}^{2975}_{train-static}$, and $\mathscr{D}^{2975}_{train-dynamic}$ were used for the retrainings. $\mathscr{D}^{19130}_{train-static}$ amounts to a total of 3826 scenarios or 19130 dataframes. Each data frame consists of both image and LiDAR scans next to their ground truth labeled counterparts. For the labeled data, both datasets use the same color coding scheme from the cityscapes dataset[23]. Each scenario is an 18.5-second long sequence in which five sensor snapshots are taken with a time difference of 4 seconds between each other. The reason for this is that the model should be able to explore most locations of the town environment. Rather than using five sequential dataframes with a time difference of 0.1 seconds between each other, each scenario provides more variety in surroundings when setting a higher time difference.

$\mathscr{D}^{2975}_{train-static}$ is a subset of $\mathscr{D}^{19130}_{train-static}$ where only 2975 images are available for training. This dataset matches the number of cityscapes[23] data frames.

The probability that the false detection of an anomaly is caused by a domain shift in weather or environment should be kept to a minimum. The simulation environment of CARLA[27] allows to replicate all environments for both training and evaluation datasets. Thus the training data and evaluation data only differ in the anomalous object present in the scenarios of the evaluation dataset. All three datasets $\mathscr{D}^{2975}_{train-static}$, $\mathscr{D}^{19130}_{train-static}$ and $\mathscr{D}^{2975}_{train-dynamic}$ therefore only include environments that also appear in the evaluation dataset $\mathscr{D}_{anomaly}$. Only CARLA towns 1-7, as well as towns 9 and 10, are used for the data generation of the training set. Similarly, only those weather conditions that also appear in the evaluation dataset are activated for the training data generation.

$\mathscr{D}_{anomaly}$ does not contain any dynamic actors, such as other vehicles or passengers, in its scenarios. To minimize the variety between the evaluation and training dataset apart from the anomalous instance, $\mathscr{D}^{19130}_{train-static}$, and $\mathscr{D}^{2975}_{train-static}$ respectively, do not include any other traffic participants.

However, as Nayal et al. describe in their paper[48], RbA thrives in detecting anomalies when they are rejected by all queries. I concluded that RbA may benefit from a higher number of trained queries. Therefore a training on $\mathscr{D}^{2975}_{train-dynamic}$ where RbA is able to refine its queries specialized on detecting dynamic actors like cars or passengers was included. Although $\mathscr{D}^{2975}_{train-dynamic}$ deviates from the normality conditions in environment settings that the other training dataset and the evaluation dataset share, the environment settings are kept the same. The dynamic actors that may appear in this dataset include the classes car, bus, truck, bicycle, motorcycle, rider, and passenger.

The environment settings for all towns that the training dataset generated included the activation of red traffic lights. This resulted in a higher percentage of training images where the ego vehicle is standing at the same spot in front of a red traffic light for multiple frames. The lower variety in surroundings that the images represent may have resulted in a bias for the method's performance. In retrospect, the training dataset could have been adjusted such that in all scenarios, the ego vehicle would not be stopped by traffic lights.

5.2 Training

Trainings were carried out as identically as possible to the training processes described in the respective papers of the anomaly detection methods. For the LiDAR method, the predictive distribution calibration training approach was selected, as this is the only training approach for ReaL where no outlier fine-tuning is utilized. Similarly, I refrained from using the fine-tuning training approach in RbA. All trainings were executed on two RTX3090 GPUs with 24GB of VRAM each.

5.2.1 Camera Training

The training configuration that Nayal et al. used to train RbA comprised a total of 90000 iterations and a learning rate of 0.0001. The batch size had to be reduced from 16 to 8 due to hardware limitations.

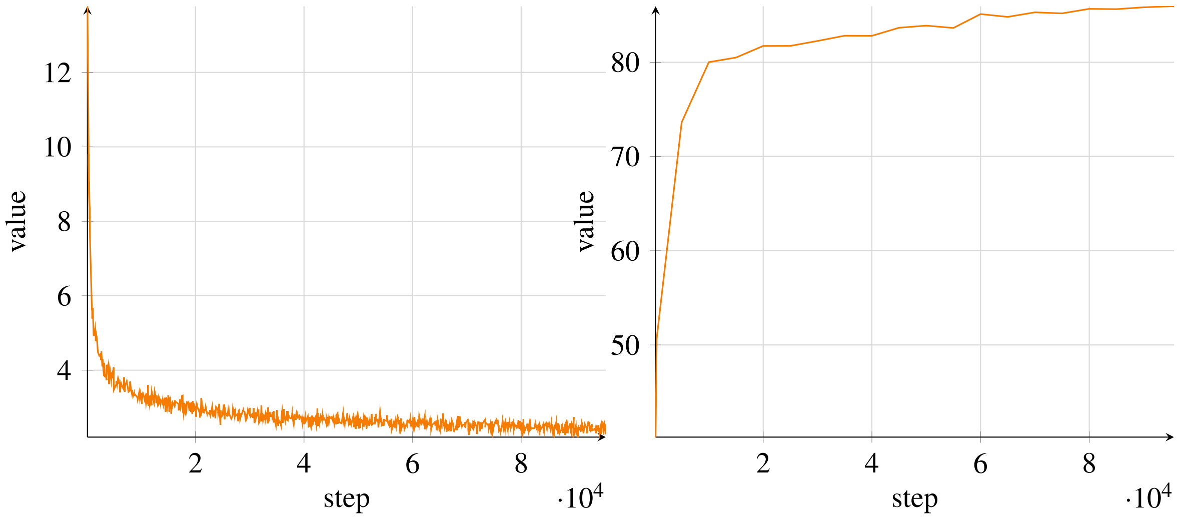
This training configuration was used for training of RbA on $\mathscr{D}^{2975}_{train-static}$, which tries to match the training described in the respective paper[48] as closely as possible. This includes matching the number of training images from cityscapes, which has a total of 2975 images in the train split and 500 images in the validation split. Hence, a portion of 3475 dataframes from the generated dataset was used.

As the size of the training dataset SemanticKITTI[5] used in ReaL is considerably larger than cityscapes[23] used for RbA, performance may differ due to the difference in exposure to training data.

Therefore, the camera method was also trained with $\mathscr{D}^{19130}_{train-static}$. This ensures that both methods are trained with the same conditions and can be compared accurately.

Training on $\mathscr{D}^{2975}_{train-dynamic}$ uses, similar as training on $\mathscr{D}^{2975}_{train-static}$, the same training configurations that were proposed in the paper.

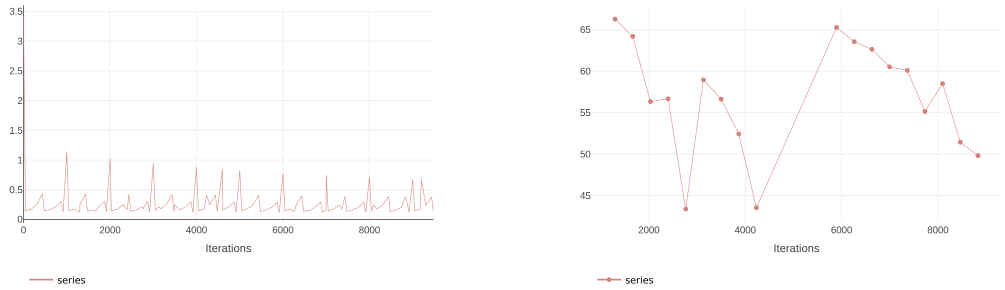
For image classification tasks, it is usually common to initialize trainings from a pretrained checkpoint. The authors of RbA used multiple checkpoints of Mask2Former that were already



(a) loss function for RbA retraining on $\mathscr{D}^{2975}_{train-static}$

(b) IoU for RbA retraining $\mathscr{D}^{2975}_{train-static}$

Figure 5.1: Training of RbA on $\mathscr{D}^{2975}_{train-static}$



(a) Total Loss

(b) mIoU on evaluation split during training

Figure 5.2: ReaL[17] was trained on $\mathscr{D}^{19130}_{train-static}$. The unusual loss function may be attributed to the implementation with a different spconv[58] version

trained on ImageNet[40]. Swin-B's[45] backbone with a single decoder layer was used as it was the best-performing backbone for RbA.

5.2.2 LiDAR Training

Training parameters were taken from Cen et al.'s experiments[16]. As mentioned, this also includes matching the size of SemanticKITTI with $\mathscr{D}^{19130}_{train-static}$. ReaL uses the same backbone from the experiments on Cylinder3D[66], which is extracted from the spconv library[58]. As the authors did not provide a description of the environment they used for their trainings, the reproduction of the experiments proved to be difficult. Notably, significant modifications in recent versions of the spconv library introduced unanticipated alterations in the model architecture. Consequently, these unexpected changes may have accounted for the abnormal curve in the loss function (figure 5.2) and the irregular anomaly predictions observable in the retrained ReaL model (see chapter 6.3).

6 Evaluation

The primary function of anomaly detection methods for autonomous driving is the reduction of risks that could arise from unexpected scenarios. If an anomaly detection method fails to detect an unknown instance, there could be dangerous consequences for the passengers. On the other hand, if an anomaly detection method falsely detects a common instance as an anomaly, a hazardous traffic situation could arise as well, i.e., the autonomous vehicle could attempt to execute an unnecessarily dangerous dodging maneuver that may endanger its passengers. Hence, it is crucial for anomaly detection methods to not excessively label anomalies in commonly occuring situations.

This chapter will discuss evaluation results on the AnoVox benchmark for the anomaly detection methods.

Given the constraints set by the adverse detection results of both methods and the odd prediction behavior of ReaL[17] the focus of this chapter will shift to the following two aspects:

- How does the performance compare between results on the voxel representation and the sensor's own representation?
- How do parameter changes in the voxel transformation method affect the results in the voxel representation?

6.1 Evaluation Metrics

Evidently, the anomaly detection method needs to have a strong ability to correctly identify anomalous and non-anomalous scenarios. Due to the heavy cost of false alarms, the number of false positives should be as minimal as possible.

To measure these characteristics, the following metrics were chosen for the evaluation: **AuPRC** (Area under Precision-Recall Curve) is the most common metric found in anomaly detection evaluation. Anomaly Detection methods have a mainly safety-critical aspect for autonomous driving. Thus, it is of great interest to measure the amount of false positive predictions, as a high count could lead to potentially dangerous traffic situations.

Since anomaly detection can be seen as a binary classification problem and anomalous objects will almost always cover a smaller region than the non-anomalous region, the AuPRC metric is well-suited as the main anomaly detection metric.

Some anomaly detection literature[9] refer to average precision (AP) as their main metric for testing the performance of their anomaly detection method. It should be mentioned that the average precision is equal to calculating the area under the precision-recall curve.

For further inspection of the safety-critical aspect of anomaly detection methods, the **FPR95** (False Positive Rate 95%) is another beneficial metric to measure the number of false positives. It shows percentage-wise how many false positives are needed to reach a total of 95% true positives.

Next to the AuPRC metric, the **AuROC** (area under the Receiver Operating Characteristic Curve) is the most widely used metric in anomaly detection[48, 3]. Although the AuROC metric's usefulness for the evaluation of anomaly detection methods is heavily discussed in literature[18, 39], it was still employed for comparison as many papers on anomaly detection methods still use it.

The scenarios in the AnoVox dataset have a substantial amount of frames where the anomaly is not in view sight. Consequently, the evaluation should assess the performance for data where no true positive predictions are to be made. Next to the false positive rate the true negative rate should also be considered for evaluating the overall performance on non-positive data. The **Specificity** is a metric that considers both false positives and true negatives. As mentioned, the specificity will only be mainly interesting for the evaluation set in which both normal and anomalous images are evaluated.

The AnoVox evaluation dataset contains multiple scenarios where small anomalies are spawned. Chan et al.[18] propose a series of metrics that specialize in evaluating the performance of anomaly detection methods on small anomalies. These include the following metrics:

Next to the AuPRC, the Positive Predictive Value (PPV) was included to further evaluate the precision of the model. The positive predictive value is equivalent to the precision score.

As the final metric, the **F1** score was selected as it rounds up false positives, true positives, and false negatives into one metric and combines the Receiver Operating Characteristic Curve with the Average Precision score.

AuPRC, FPR95, and AuROC are calculated threshold independently. All other metrics require a threshold at which a voxel is classified as anomalous or non-anomalous. For the F1 score and PPV metrics, an average over multiple thresholds was chosen $\tau \in \{0.25, 0.30, ..., 0.75\}$ that was also used for the component-wise metrics in the SegmentMeIfYouCan benchmark. The specificity metric seems to provide the most insight into true negative detection when the maximum of the precision-recall curve is used as the threshold.

6.2 Preprocessing for the Evaluation

To make the comparison as fair as possible, only the voxels within reach of the sensors were evaluated. This means that only voxels in front of the camera are relevant, as the environment behind the ego vehicle cannot be detected. Given a voxel grid from the prediction and a voxel grid from the ground truth, the relevant voxels for evaluation are extracted by intersecting both voxel grids.

Transferring a camera image into a 3D space like a voxel grid requires additional information on the depth for each pixel of the image. In a real-world application, a transfer from raw camera data into a voxel space will be aided by a depth estimator module that calculates absolute depth for each pixel. However, to study the camera detection method as a single component, the ground truth depth image that the AnoVox dataset provides was used so that inaccuracies in depth measurement don't influence the results negatively. For a voxel grid prediction where depth measurements are

	Anomalies Only						Normality Included				
	AuPRC	FPR_{95}	AuROC	Specificity	PPV	F_1	AuPRC	FPR_{95}	AuROC	Specificity	PPV
RbA $\mathscr{D}^{2975}_{train-static}$	1.4	95.7	73.9	99.8	2.6	4.7	0.5	95.8	74.0	99.9	0.8
RbA $\mathscr{D}^{19130}_{train-static}$	1.0	95.2	74.0	99.8	1.9	3.4	0.2	95.6	74.2	99.9	0.5
RbA $\mathscr{D}^{2975}_{train-dynamic}$	0.7	100	57.3	99.8	1.4	2.6	0.2	100	57.6	99.9	0.4
ReaL $\mathscr{D}^{19130}_{train-static}$	0	73.9	57.0	99.7	0	0	0.2	74.2	57.0	99.7	0

Table 6.1: Evaluation Results on AnoVox. The evaluation benchmark was split into a subset where the anomaly is detectable in every frame. Furthermore, the entire set of frames that includes frames where the anomaly cannot be seen, was used as well. Noticably, a heavy drop in precision based metrics can be observed when comparing performance on the voxel volume to the evaluation on the initial sensor. Moreover, performance drops off once frames with no positives is mixed into the evaluation benchmark.

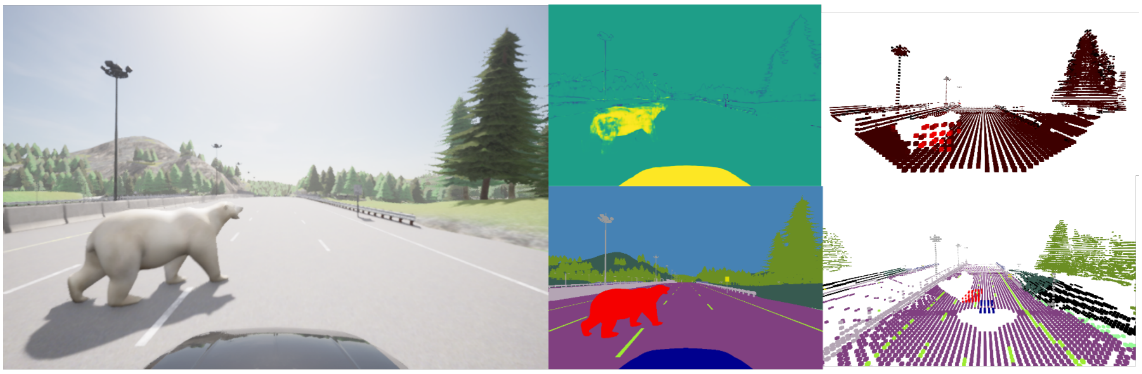


Figure 6.1: Prediction of frame 7256. The values in the precision-based metrics drop as the voxel grid has a higher class imbalance than in the sensor data

extracted from a depth estimator, an intersection between the predicted voxel grid and ground truth voxel grid would not suffice. However, as all ground truth voxel grids in the AnoVox evaluation dataset were created from the ground truth depth images, the grid intersection will not result in additional voxels from the prediction being left over.

RbA was not able to correctly label the engine hood as non-anomalous in most cases, which resulted in strong distortions of the precision scores in the evaluation. Because of this, the engine hood was masked from all images as background regions that can be neglected in the evaluation.

6.3 Evaluation Results

Results are provided for the following models:

RbA trained on $\mathscr{D}^{2975}_{train-static}$. No dynamic actors appear in this training split which makes this split deviate from $\mathscr{D}_{anomaly}$ in the appearance of anomalous instances. The training parameters match the configurations used in Nayal et al.'s experiments[48].

RbA trained on $\mathscr{D}^{19130}_{train-static}$ Similarly, no dynamic actors appear in this training split. Training parameters match the configurations from Cen et al.'s experiments[17].

RbA trained on $\mathscr{D}^{2975}_{train-dynamic}$ where dynamic actors appear. The training parameters match Nayal et al.'s training configuration.

ReaL trained on $\mathscr{D}^{19130}_{train-static}$ where dynamic actors appear. The training configurations were taken from Cen et al.'s experiments.

The evaluation dataset $\mathscr{D}_{anomaly}$ consists of ten scenarios where an anomaly appears in varying sizes each.

During the evaluation, a potential error was found in the intersection operation for the anomaly score voxel grids and the ground truth voxel grids. Whilst the results for the RbA model trained on $\mathscr{D}^{2975}_{train-static}$ with a voxel resolution 0.5m³ only deviates by less than 0.75 percent for all metrics when compared to the new results, further analysis should be conducted. Due to the time limitation of this thesis, results could not be provided with the overworked evaluation script.

All trained models reached adverse results on the voxelized $\mathscr{D}_{anomaly}$ evaluation split as can be seen in table 6.1. The AuPRC value does not reach over 2% in any models. This implies that all models struggled to contain their positive predictions to the anomalous regions in the voxel volume. The FPR$_{95}$ values furthermore confirm that the models predicted an overly high amount of false positives. Whilst the AuROC values over 50% suggests that the models were able to detect anomalous regions in some frames, the F$_1$ and PPV scores further underline the models' weak ability to distinctly classify anomalies and to discriminate from non-anomalous regions.

The $\mathscr{D}^{2975}_{train-dynamic}$ trained RbA model's expected higher recall performance stayed out and turned out to perform significantly worse than its counterpart models retrained on the static dataset.

Anomaly score predictions from the retrained ReaL model seemed to be randomized and did not appear to formulate any attempts to segment the data into regions. Consequently, a direct comparison between the camera and LiDAR method was neglected.

The evaluation on $\mathscr{D}_{anomaly}$ was separated into multiple splits, where the performance is explored in the context of a specific characteristic added to the evaluation dataset. This includes the addition of frames with no positives, where no anomalies are detectable, as well as the evaluation results in correlation to the size of the anomalous instance. Moreover, evaluation results in the methods' respective sensor domains are provided for comparison.

6.3.1 Results with Normality

To analyze the models' performance on data where no positive values are contained, all models were evaluated on $\mathscr{D}^{norm}_{anomaly}$ where frames with no anomalous regions were included (6.1). $\mathscr{D}^{norm}_{anomaly}$ as a whole consists of 1850 frames, from which 1510 frames do not contain any anomalous regions in the images. The remaining 340 frames are the same as the frames in the original $\mathscr{D}_{anomaly}$. The drop in all precision-based metrics is plausible as any positive predictions for the normal frames will be classified as false positives and reduce the overall precision performance. The high specificity values indicate that the models are able to correctly identify non-anomalous regions as negatives. This, however, should not be interpreted as a measurement of the models' detection ability and only be taken as an additional measurement to the precision- and recall-based metrics. Since $\mathscr{D}^{norm}_{anomaly}$ is highly imbalanced in the ratio of positives and negatives, correctly identifying negatives is no testament to a strong detection ability of positives.

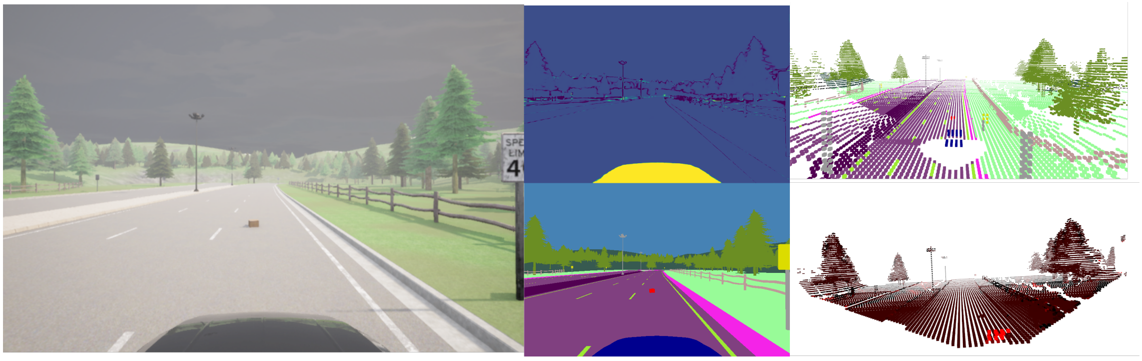


Figure 6.2: Frame 7465 shows a small anomaly in the shape of a card box. Both RbA and ReaL struggled to correctly identify any of the small anomalies that were presented in the evaluation dataset.

	Anomalies Only			Normality Included		
	AuPRC	FPR_{95}	AuROC	AuPRC	FPR_{95}	AuROC
RbA $\mathscr{D}^{2975}_{train-static}$	21.6	99.7	64.6	11.8	99.7	65.5
RbA $\mathscr{D}^{19130}_{train-static}$	25.0	98.3	72.6	13.65	98.38	73.70
RbA $\mathscr{D}^{2975}_{train-dynamic}$	8.1	38.2	38.2	3.45	99.70	39.08
ReaL $\mathscr{D}^{19130}_{train-static}$	0.8	88.9	31.9	0.5	88.4	32.1

Table 6.2: Scores for RbA and ReaL in their respective sensor domain.

6.3.2 Detection Ability in relation to the size of the anomaly

Due to the anomaly size attribute provided in the AnoVox benchmark for every scenario, the models' performances can be showcased for each possible size of the anomalous instance. The frames included in this evaluation evidently have to contain a detectable anomalous region for the sensor.

Expectedly, the smaller the unknown instance is, the harder it is for the model to correctly label it as an anomaly All results can be seen in table 6.3. Moreover, all small anomalies in $\mathscr{D}_{anomaly}$ are cubic-shaped. As cuboids frequently appear as static or dynamic objects in the training dataset as cardboxes or mailboxes the model may have learned to classify such as known objects. It is likely that this additionally contributed to the weaker detection performances for small objects as seen in figure 6.2.

6.3.3 Evaluation on Sensors

Performance of the models in their own respective data representation is provided for the main metrics AuPRC, FPR_{95} and AuROC in table 6.2.

Noticeably, RbA shows a much higher area under precision-recall curve value when compared to its performance on voxel grids.

	BIG					**MEDIUM**					**SMALL**				
	AuPRC	FPR_{95}	AuROC	PPV	F_1	AuPRC	FPR_{95}	AuROC	PPV	F_1	AuPRC	FPR_{95}	AuROC	PPV	F_1
RbA $\mathscr{D}^{2975}_{train-static}$	18.8	92.9	80.7	11.2	14.7	0.6	100	68.4	1.0	1.9	0.01	74.1	50.8	0.02	0.04
RbA $\mathscr{D}^{19130}_{train-static}$	8.4	91.2	82.3	8.0	10.3	0.6	100	67.0	0.9	1.8	0.01	61.27	56.7	0.03	0.06
RbA $\mathscr{D}^{2975}_{train-dynamic}$	2.3	100	54.9	2.7	3.7	0.6	100	60.5	0.02679	0.03705	0.01	100	53.2	0	0
Lidar $\mathscr{D}^{19130}_{train-static}$	0.2	75.5	54.1	0	0	0.00289	0.75591	0.54194	0	0	0.1	28.0	87.6	0	0

Table 6.3: Anomaly Detection Results ordered after size of the anomaly. Expectedly, bigger anomalies were likelier to be detected. Deriving from the AuROC scores for small anomalies, the methods were not able to differentiate between anomalous and non-anomalous regions when encountering small anomalies.

	AuPRC	FPR_{95}	AuROC	Specificity	PPV	F_1
RbA $\mathscr{D}^{2975}_{train-static}$	1.3	95.7	73.7	99.7	2.4	4.3
RbA $\mathscr{D}^{19130}_{train-static}$	1.0	95.2	74.0	99.8	1.9	3.4
RbA $\mathscr{D}^{2975}_{train-dynamic}$	0.6	100	57.4	99.8	1.4	2.6
Lidar $\mathscr{D}^{19130}_{train-static}$	0.2	73.9	57.0	99.7	0	0

Table 6.4: Evaluation scores for mean function.

6.3.4 Comparison of different Voxel Functions

As discussed in 4.4.1, the results for voxelization scores were provided where the anomaly score of a voxel is calculated by the mean of the scores of all points located inside its region. Changing the voxelization function does not significantly impact the evaluation results in the voxel space as can be seen in table 6.4.

6.3.5 Comparison of Results between Image and Voxel Representation

RbA achieves an average precision of 25.0%, an AuROC score of 72.6%, and a FPR_{95} score of 98.3% on $\mathscr{D}_{anomaly}$.

The most plausible explanation for the significant decrease in precision is the higher class imbalance in the voxel data that results from the voxelization. The ratio of anomalous pixels to the total amount of pixels equals 4.159% whilst the same ratio falls down to 0.14% when evaluating on voxels. The high FPR_{95} score already indicates that all models struggled with containing false positive predictions to a minimum. As the amount of positives in the ground truth decreases proportionally, the model has higher chances of labeling voxels as anomalies that are not positives.

Frame 7256 in the AnoVox dataset may be able to emphasize the effect the change in class imbalance has on the scores. Despite the lower FPR_{95} score of 16.27% in the voxel grid for voxels, the higher class imbalance led to a drop from 70.56% to 50.92% in average precision.

The low FPR_{95} score in frame 7256, however, may be misleading. For most frames, like in Frame 5497 seen in figure 6.3, the FPR_{95} score equaled 100%.

Another factor that most likely impacted the results was an occasional absence of positive values in the ground truth voxel grid even though the anomaly was detectable in the image. This usually occurs with small anomalies where the voxelization function may ignore the anomaly labels from

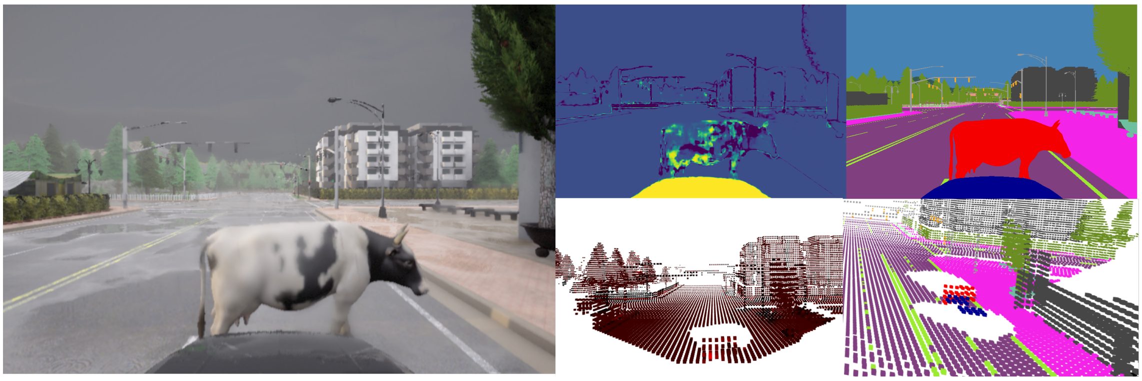


Figure 6.3: Frame 5497 of the AnoVox dataset. Similar to other frames, the FPR_{95} score for this frame equaled 100%.

Voxel Resolution 0.2					
	AuPRC	FPR_{95}	AuROC	PPV	F_1
RbA $\mathscr{D}^{2975}_{train-static}$	1.8	100	70.0	3.3	5.9
RbA $\mathscr{D}^{19130}_{train-static}$	1.5	100	71.8	2.5	4.5
RbA $\mathscr{D}^{2975}_{train-dynamic}$	0.9	100	55.7	2.0	3.6
Lidar $\mathscr{D}^{19130}_{train-static}$	0.1	96.3	32.2	0.0	0.0

Table 6.5: Results on a Voxel Resolution of 0.2m. Precision performance was slightly better than on a voxel resolution of 0.5m. This further indicates that the drop in performance can be traced back to a higher class imbalance, as the ratio of positives to the total number of voxels increased from 0.14% to 0.19% here.

the pseudo point cloud as they are not the closest to the center point of a voxel. This factor only affected results in 6.3.1 since a data frame was only added to $\mathscr{D}_{anomaly}$ if both sensor data and voxel data contained anomalies to be detected.

The decision to leave the engine hood viewable in $\mathscr{D}_{anomaly}$ further complicated the evaluation process. Despite masking the engine hood from the image, misalignments of its shape can be found in some frames. This led to unmasked pixels from the engine hood with high anomaly scores being transferred into the prediction voxel grid and further distorting the precision of the model.

Whilst the higher class imbalance is the likeliest explanation for the noticeable decrease in precision performance, there may be other factors that could have been overlooked. Further research would be needed to confirm if a drop from 25% to less than 2% in average precision can be solely attributed to the decrease in positive value ratio in the data.

6.3.6 Impact of Voxel Resolution on Performance

Changing the resolution of the voxels slightly improves the precision metrics of the results when compared to the results on the larger voxel size of 0.5m³. Results were provided for evaluation with a voxel resolution of 0.2m³ in table 6.5. This furthermore indicates that the higher class imbalance in the data contributes to the significant drop in precision.

7 Conclusion

Whilst the field of anomaly detection in autonomous driving has experienced steady advancement over the past years, paralleling the rapid progress of the general research field of computer vision, there remain some relevant aspects within the autonomous driving context that present research tends to disregard. Anomaly Detection on the Object Level[13] can be broadly classified as a subproblem of image segmentation, and it is therefore expected that most research will draw inspiration from the classical evaluation approaches in the image segmentation domain. However, since nowadays' autonomous vehicles usually have to process multiple sensor information and unify the sensor inputs, it is of key interest to evaluate a method's performance in the common state representation. As the evaluation results can change due to higher class imbalance or inaccurate depth estimations in the common state representation, additional parameters like the remapping of anomaly scores into the new 3D space or, specifically to voxel grids, the resolution of each voxel has to be taken into account.

The results presented in this thesis emphasize the importance of multimodal evaluation. The substantial decrease in precision performance during the transformation of detection results from the camera domain into voxels, as observed in the evaluation, indicates that there are further factors that have to be considered when evaluating methods on a common representation such as a voxel grid. For example, slight changes in results were observed when reconfiguring parameters such as the voxel resolution or the function for determining the anomaly score for a single voxel. It is however assumed that the decrease in precision performance is caused by a significantly increasing class imbalance compared to the image representation of the data.

The heavy focus on image-based anomaly detection methods in autonomous driving in the literature evidently leads to anomaly detection research falling behind for other sensor domains. Research into LiDAR anomaly detection proved to be more challenging than for camera-based methods. The general absence of standardized evaluation benchmarks in the LiDAR domain forces researchers to evaluate their methods on custom benchmarks, which complicates performance comparison. It is evident that research for other sensor domains can be accelerated with releases of standardized evaluation benchmarks where methods can be summarized and compared more accurately.

7.1 Outlook

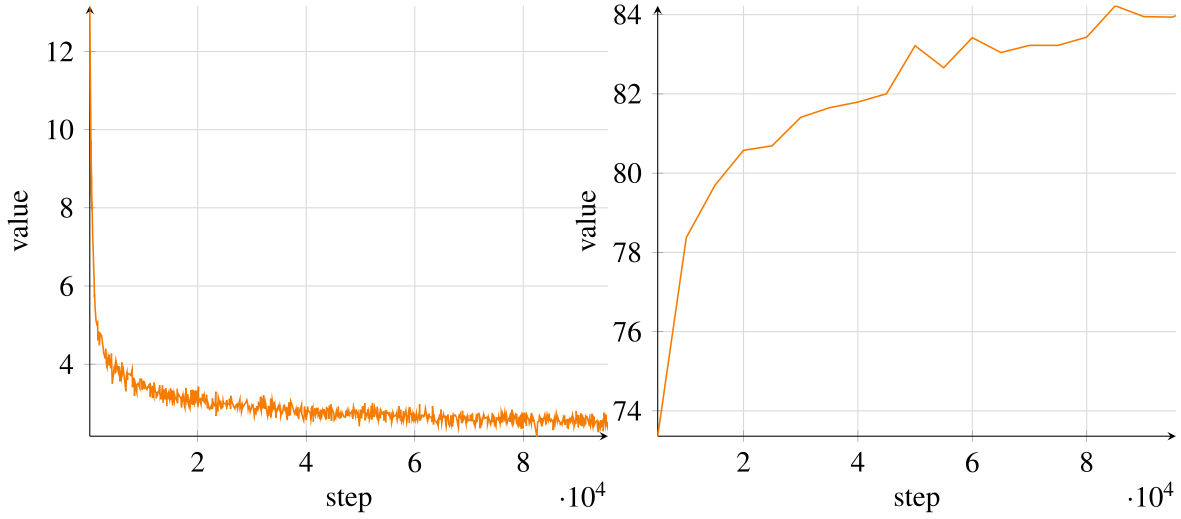
The results in this thesis leave some research aspects open that future work could build upon.

The comparison between the camera and LiDAR anomaly detection forcibly fell short due to unexpected difficulties arising in the implementation of ReaL (5.2.2). It would be of interest for future research to evaluate the performance of methods from different sensor domains side-by-side with their own respective strengths and weaknesses.

In practice, an autonomous vehicle deployed on the street would utilize a depth estimator module to transform its image features into a fused representation. Depth estimators, however, have limitations in accuracy and are significantly outperformed by specialized range sensors such as LiDAR. Thus, a comparison between a LiDAR-based and a camera-based anomaly detection method extended with a depth estimator would be more informative from an application viewpoint. Future research could analyze the effects that would appear in the voxel representation due to inaccuracies of the depth estimator module. In the context of this thesis, the distorted depth scale of the depth predictions could possibly be rearranged by retraining the depth estimator on images taken in CARLA[27]. One would also have to consider different evaluation strategies to handle falsely occupied voxels caused by wrong predictions of the depth estimator.

Whilst the increase in class imbalance in the voxel data is the most likely explanation for the drop in precision, there may be additional factors that might have affected the results. Further research could provide insights into other parameter configurations for the voxel transformation that may impact the shift in evaluation outcomes.

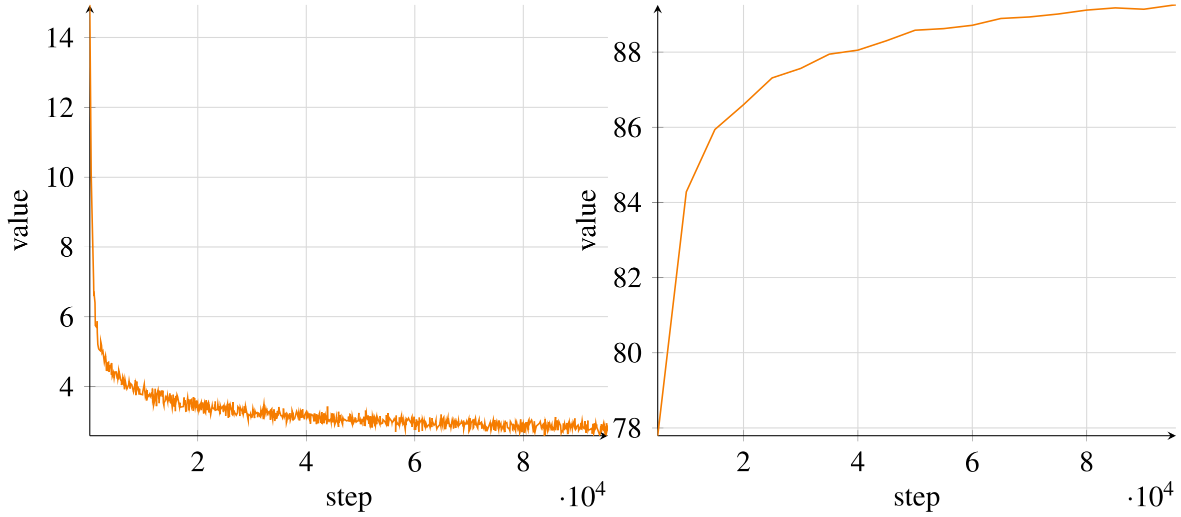
A Appendix



(a) loss function for RbA retraining on $\mathscr{D}^{19130}_{train-static}$

(b) IoU for RbA retraining on $\mathscr{D}^{19130}_{train-static}$

Figure A.1: Training of RbA on $\mathscr{D}^{19130}_{train-static}$



(a) loss function for RbA retraining on $\mathscr{D}^{2975}_{train-dynamic}$

(b) IoU for RbA retraining on $\mathscr{D}^{2975}_{train-dynamic}$

Figure A.2: Training of RbA on $\mathscr{D}^{2975}_{train-dynamic}$

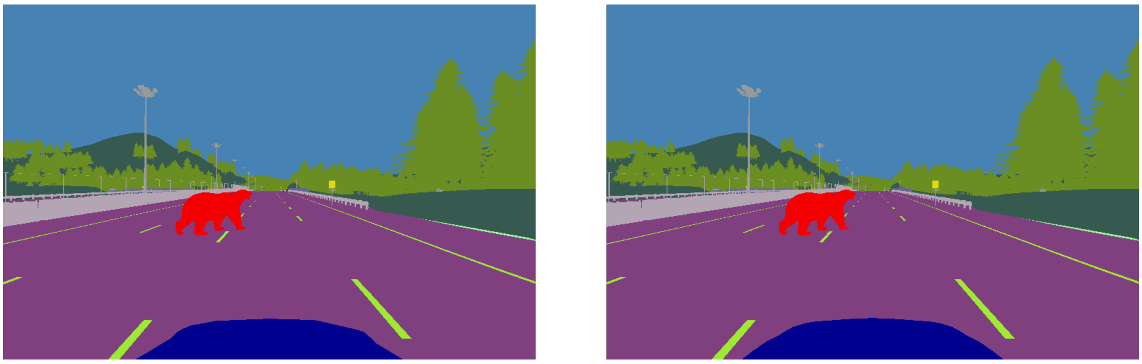


Figure A.3: Comparison between Frame 7235 and 7236 show the misalignment in the engine hood. This resulted in voxels appearing in the prediction voxel grid with high anomaly scores that originate from the predictions on the engine hood.

B List of Figures

C List of Tables

D Bibliography

[1] J. Ackermann, C. Sakaridis, and F. Yu. Maskomaly:zero-shot mask anomaly segmentation. *arXiv preprint:2305.16972*, 2023.

[2] K. M. A. Alheeti, A. Alzahrani, and D. Al Dosary. Lidar spoofing attack detection in autonomous vehicles. In *2022 IEEE International Conference on Consumer Electronics (ICCE)*, 2022.

[3] A. Alliegro, F. C. Borlino, and T. Tommasi. 3dos: Towards 3d open set learning – benchmarking and understanding semantic novelty detection on point clouds. *arXiv preprint:2207.11554*, 2023.

[4] E. Ashok. Occupancy networks. https://www.youtube.com/watch?v=jPCV4GKX9Dw, 2022.

[5] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[6] Behley, Jens. SemanticKITTI: Open World Lidar Instance Segmentation®. `https://codalab.lisn.upsaclay.fr/competitions/2183`, 2022. Accessed: 2023-11-11.

[7] V. Besnier, A. Bursuc, D. Picard, and A. Briot. Triggering failures: Out-of-distribution detection by learning from local adversarial attacks in semantic segmentation. *arXiv preprint:2108.01634*, 2021.

[8] S. F. Bhat, R. Birkl, D. Wofk, P. Wonka, and M. Müller. ZoeDepth: Zero-shot transfer by combining relative and metric depth, 2023.

[9] H. Blum, P.-E. Sarlin, J. Nieto, R. Siegwart, and C. Cadena. The fishyscapes benchmark: Measuring blind spots in semantic segmentation. *arXiv preprint:1904.03215*, 2021.

[10] D. Bogdoll, I. Hamdard, L. Roessler, F. Wang, M. Bayram, and F. Geisler. Anovox. `https://doi.org/10.5281/zenodo.8171712`, 2023.

[11] D. Bogdoll, M. Nitsche, and J. M. Zöllner. Anomaly detection in autonomous driving: A survey. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2022.

[12] D. Bogdoll, S. Uhlemeyer, K. Kowol, and J. M. Zöllner. Perception datasets for anomaly detection in autonomous driving: A survey. *arXiv preprint:2302.02790*, 2023.

[13] J. Breitenstein, J.-A. Termöhlen, D. Lipinski, and T. Fingscheidt. Systematization of corner cases for visual perception in automated driving. In *2020 IEEE Intelligent Vehicles Symposium (IV)*.

[14] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[15] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. *arXiv preprint:2005.12872*, 2020.

[16] J. Cen, P. Yun, J. Cai, M. Y. Wang, and M. Liu. Open-set 3d object detection. In *2021 International Conference on 3D Vision (3DV)*, 2021.

[17] J. Cen, P. Yun, S. Zhang, J. Cai, D. Luan, M. Y. Wang, M. Liu, and M. Tang. Open-world semantic segmentation for LIDAR point clouds. *arXiv preprint:2207.01452*, 2022.

[18] R. Chan, K. Lis, S. Uhlemeyer, H. Blum, S. Honari, R. Siegwart, P. Fua, M. Salzmann, and M. Rottmann. SegmentMeIfYouCan: A benchmark for anomaly segmentation. *arXiv preprint:2104.14812*, 2021.

[19] R. Chan, M. Rottmann, and H. Gottschalk. Entropy maximization and meta classification for out-of-distribution detection in semantic segmentation. (arXiv:2012.06575), 2021.

[20] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 2009.

[21] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar. Masked-attention mask transformer for universal image segmentation. *arXiv preprint:2112.01527*, 2022.

[22] B. Cheng, A. G. Schwing, and A. Kirillov. Per-pixel classification is not all you need for semantic segmentation. *arXiv preprint:2107.06278*, 2021.

[23] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[24] Cordts, Marius. Cityscapes TrainID relabeling script. `https://github.com/mcordts/cityscapesScripts`, 2016. Accessed: 2023-11-20.

[25] W. Deng, K. Huang, Q. Yu, H. Lu, Z. Zheng, and X. Chen. ElC-OIS: Ellipsoidal clustering for open-world instance segmentation on LiDAR data. *arXiv preprint:2303.04351*, 2023.

[26] G. Di Biase, H. Blum, R. Siegwart, and C. Cadena. Pixel-wise anomaly detection in complex driving scenes. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[27] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, 2017.

[28] A. Floris, L. Frittoli, D. Carrera, and G. Boracchi. Composite layers for deep anomaly detection on 3d point clouds. *arXiv preprint:2209.11796*, 2022.

[29] S. F. Frisken and R. N. Perry. Simple and efficient traversal methods for quadtrees and octrees. *Journal of Graphic Tools,*, 2002.

[30] S. Galesso, M. Argus, and T. Brox. Far away in the deep space: Dense nearest-neighbor-based out-of-distribution detection. 2023.

[31] S. Gebhardt, E. Payzer, L. Salemann, A. Fettinger, E. Rotenberg, and C. Seher. Polygons, point-clouds, and voxels, a comparison of high-fidelity terrain representations.

[32] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[33] E. Goan and C. Fookes. Bayesian neural networks: An introduction and survey. 2020.

[34] M. Grcić, P. Bevandić, Z. Kalafatić, and S. Šegvić. Dense out-of-distribution detection by robust learning on synthetic negative data, 2021.

[35] M. Grcić, P. Bevandić, and S. Šegvić. DenseHybrid: Hybrid anomaly detection for dense open-set recognition. In S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, editors, *Computer Vision – ECCV 2022*, 2022.

[36] M. Grcić, J. Šarić, and S. Šegvić. On advantages of mask-level recognition for outlier-aware segmentation. *arXiv preprint:2301.03407*, (arXiv:2301.03407), 2023.

[37] M. Higgins, D. Jha, and D. Wallom. Spatial-temporal anomaly detection for sensor attacks in autonomous vehicles. (arXiv:2212.07757), 2022.

[38] P. Hu, D. Held, and D. Ramanan. Learning to optimally segment point clouds. *arXiv preprint:1912.04976*, 2019.

[39] G. Humblot-Renaux, S. Escalera, and T. B. Moeslund. Beyond AUROC & co. for evaluating out-of-distribution detection performance, 2023.

[40] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

[41] S. Laine and T. Karras. Efficient sparse voxel octrees. In *Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games*, 2010.

[42] J. Li and Q. Dong. Open-set semantic segmentation for point clouds via adversarial prototype framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[43] K. Lis, K. Nakka, P. Fua, and M. Salzmann. Detecting the unexpected via image resynthesis, 2019.

[44] Y. Liu, C. Ding, Y. Tian, G. Pang, V. Belagiannis, I. Reid, and G. Carneiro. Residual pattern learning for pixel-wise out-of-distribution detection in semantic segmentation. *arXiv preprint:2211.14512*, 2023.

[45] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[46] M. Masuda, R. Hachiuma, R. Fujii, H. Saito, and Y. Sekikawa. Toward unsupervised 3d point cloud anomaly detection using variational autoencoder. 2021.

[47] M. Najibi, J. Ji, Y. Zhou, C. R. Qi, X. Yan, S. Ettinger, and D. Anguelov. Motion inspired unsupervised perception and prediction in autonomous driving. In *Computer Vision – ECCV 2022*. Springer Nature Switzerland, 2022.

[48] N. Nayal, M. Yavuz, J. F. Henriques, and F. Güney. RbA: Segmenting unknown regions rejected by all. *arXiv preprint:2211.14293*, (arXiv:2211.14293), 2023.

[49] L. Nunes, X. Chen, R. Marcuzzi, A. Osep, L. Leal-Taixe, C. Stachniss, and J. Behley. Unsupervised class-agnostic instance segmentation of 3d LiDAR data for autonomous vehicles. *IEEE Robotics and Automation Letters*, 2022.

[50] L. Nunes, R. Marcuzzi, X. Chen, J. Behley, and C. Stachniss. SegContrast: 3d point cloud feature representation learning through self-supervised segment discrimination. *IEEE Robotics and Automation Letters*, 2023.

[51] P. Pinggera, S. Ramos, S. Gehrig, U. Franke, C. Rother, and R. Mester. Lost and found: Detecting small road hazards for self-driving vehicles. *arXiv preprint:1609.04653*, 2016.

[52] A. Piroli, V. Dallabetta, J. Kopp, M. Walessa, D. Meissner, and K. Dietmayer. Energy-based detection of adverse weather effects in LiDAR data. 2023.

[53] A. Piroli, V. Dallabetta, J. Kopp, M. Walessa, D. Meissner, and K. Dietmayer. LS-VOS: Identifying outliers in 3d object detections using latent space virtual outlier synthesis, 2023.

[54] F. Poux and R. Billen. Voxel-based 3d point cloud semantic segmentation: Unsupervised geometric and relationship featuring vs deep learning methods. *ISPRS International Journal of Geo-Information*, 2019.

[55] S. N. Rai, F. Cermelli, D. Fontanel, C. Masone, and B. Caputo. Unmasking anomalies in road-scene segmentation. *arXiv preprint:2307.13316*, 2023.

[56] T. Saito and M. Rehmsmeier. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE*, 2015.

[57] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint:1312.6034v2*, 2013.

[58] Spconv and Contributors. Spconv: Spatially sparse convolution library. `https://github.com/traveller59/spconv`, 2022.

[59] P. K. Vinodkumar, D. Karabulut, E. Avots, C. Ozcinar, and G. Anbarjafari. A survey on deep learning based segmentation, detection and classification for 3d point clouds. *Entropy*, 2023.

[60] Y. Wang, J. Peng, J. Zhang, R. Yi, Y. Wang, and C. Wang. Multimodal industrial anomaly detection via hybrid fusion. *arXiv preprint:2303.00601*, 2023.

[61] K. Wong, S. Wang, M. Ren, M. Liang, and R. Urtasun. Identifying unknown instances for autonomous driving. In *Proceedings of the Conference on Robot Learning*, 2020.

[62] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[63] Y. Xu, X. Tong, and U. Stilla. Voxel-based representation of 3d point clouds: Methods, applications, and its potential use in the construction industry. *Automation in Construction, volume 126*, 2021.

[64] C. You, Z. Hau, and S. Demetriou. Temporal consistency checks to detect LiDAR spoofing attacks on autonomous vehicle perception. In *Proceedings of the 1st Workshop on Security and Privacy for Mobile AI*, 2021.

[65] Y. Zhou and O. Tuzel. VoxelNet: End-to-end learning for point cloud based 3d object detection. *arXiv preprint:1711.06396*, 2017.

[66] X. Zhu, H. Zhou, T. Wang, F. Hong, Y. Ma, W. Li, H. Li, and D. Lin. Cylindrical and asymmetrical 3d convolution networks for LiDAR segmentation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.