
SegmentMeIfYouCan: A Benchmark for Anomaly Segmentation

Robin Chan^{*1} **Krzysztof Lis^{*2}** **Svenja Uhlemeyer^{*1}** **Hermann Blum^{*3}** **Sina Honari²**

Roland Siegwart³ **Pascal Fua²** **Mathieu Salzmann²** **Matthias Rottmann¹**

Abstract

State-of-the-art semantic or instance segmentation deep neural networks (DNNs) are usually trained on a closed set of semantic classes. As such, they are ill-equipped to handle previously-unseen objects. However, detecting and localizing such objects is crucial for safety-critical applications such as perception for automated driving, especially if they appear on the road ahead. While some methods have tackled the tasks of anomalous or out-of-distribution object segmentation, progress remains slow, in large part due to the lack of solid benchmarks; existing datasets either consist of synthetic data, or suffer from label inconsistencies. In this paper, we bridge this gap by introducing the “SegmentMeIfYouCan” benchmark. Our benchmark addresses two tasks: Anomalous object segmentation, which considers any previously-unseen object category; and road obstacle segmentation, which focuses on any object on the road, may it be known or unknown. We provide two corresponding datasets together with a test suite performing an in-depth method analysis, considering both established pixel-wise performance metrics and recent component-wise ones, which are insensitive to object sizes. We empirically evaluate multiple state-of-the-art baseline methods, including several models specifically designed for anomaly / obstacle segmentation, on our datasets and on public ones, using our test suite. The anomaly and obstacle segmentation results show that our datasets contribute to the diversity and difficulty of both data landscapes.

1 Introduction

The advent of high-quality publicly-available datasets, such as Cityscapes [1], BDD100k [2], A2D2 [3] and COCO [4] has hugely contributed to the progress in semantic segmentation. However, while state-of-the-art deep neural networks (DNNs) yield outstanding performance on these datasets, they typically provide predictions for a closed set of semantic classes. Consequently, they are unable to classify an object as *none of the known categories* [5]. Instead, they tend to be overconfident in their predictions, even in the presence of previously-unseen objects [6], which precludes the use of uncertainty to identify the corresponding anomalous regions.

Nevertheless, reliability in the presence of unknown objects is key to the success of applications that have to face the diversity of the real world, *e.g.*, perception in automated driving. This has motivated the creation of benchmarks such as FishyScapes [7] or CAOS [8]. While these benchmarks have enabled interesting experiments, the limited real-world diversity in FishyScapes, the lack of a

^{*}equal contribution

¹Stochastics Group, IZMD, University of Wuppertal, Wuppertal, Germany

²Computer Vision Laboratory, EPFL, Lausanne, Switzerland

³Autonomous Systems Lab, ETHZ, Zürich, Switzerland

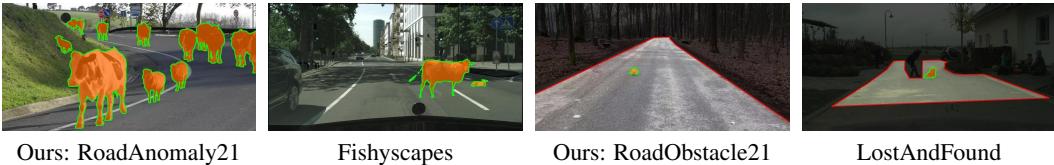


Figure 1: Comparison of images from our and existing public datasets. Anomalies / obstacles are highlighted in orange, darkened regions are excluded from the evaluation. In RoadAnomaly21, anomalies may appear everywhere in the image. In contrast to Fishscapes, where anomalous objects are synthetic, all RoadAnomaly21 images are real. In RoadObstacle21, the region of interest is restricted to the drivable area with obstacles ahead. This is comparable to LostAndFound, where the labeling, however, is not always consistent, *e.g.* children are anomalies but other humans not.

public leader board and of a benchmark suite in CAOS, and the reliance on synthetic images in both benchmarks hinder proper evaluation of and comparisons between the state-of-the-art methods.

In this paper, motivated by the limitations of existing anomaly segmentation datasets and by the emerging body of works in this direction [7, 9, 10, 11, 12, 13, 14, 15, 16], we introduce the *SegmentMeIfYouCan*² benchmark. It is accompanied with two datasets, consisting of diverse and manually annotated real images, a public leader board and an evaluation suite, providing in-depth analysis and comparisons, to facilitate the development of road anomaly segmentation methods.

Our benchmark encompasses two separate tasks. The first one consists of strict anomaly segmentation, where any previously-unseen object is considered as an anomaly. Furthermore, motivated by the observation that the boundary between known and unknown classes can sometimes be fuzzy, for instance for *car vs. van*, we introduce the task of obstacle segmentation, whose goal is to identify all objects on the road, may they be from known classes or from unknown ones.

For the anomaly track, we provide a dataset of 100 images with pixel-wise annotations over two classes (anomaly, not anomaly) and a void class, which, in analogy to Cityscapes, signals the pixels that are excluded from the evaluation. We consider any object that strictly cannot be seen in the Cityscapes data as anomalous, appearing anywhere in the image. For the obstacle track, our dataset contains 327 images with analogous annotation (obstacle, not obstacle, void), and focuses only on the road as region of interest. The focus in this track is of more practical need, *e.g.* for automated driving systems, targeting obstacles that may cause hazardous street situations, see Figure 1. All images of our datasets are publicly available for download², together with a benchmark suite that computes both established pixel-wise metrics and recent component-wise ones.

In the remainder of this paper, we first review existing anomaly detection datasets, methods and evaluation metrics in more detail. We then describe our new benchmark and provide extensive experiments comparing state-of-the-art road anomaly / obstacle segmentation methods on our datasets and on other related ones, showing the difficulty of the models on the proposed benchmarks.

2 Related Work

In this section we first review previous datasets for anomaly detection, with some of them being designed for road anomaly segmentation. Then we briefly describe some of the methods on anomaly and obstacle segmentation.

2.1 Datasets and Benchmarks

Existing methods for anomaly detection have often been evaluated on their ability to separate images from two different source distributions, such as separating MNIST from FashionMNIST [17, 18, 19], NotMNIST [19], or Omniglot [20], and separating CIFAR-10 from SVHN [18, 19, 21] or LSUN [18, 21, 22]. Such experiments can be found in many works, including [17, 18, 19, 21, 22, 23].

²<https://www.segmentmeifyoucan.com/>

Dataset	anomaly pixels	non-anomaly pixels	diverse scenes	different anomalies	dataset size	ground truth (gt) components	mean & std of gt size relative to image size
Fishscapes LostAndFound val [7]	0.23%	81.13%	12	7	373	165	$0.13\% \pm 0.23\%$
CAOS BDD-Anomaly test [8]	0.83%	81.28%	810	3	810	1231	$0.55\% \pm 1.84\%$
Ours: RoadAnomaly21 test	13.83%	82.17%	100	26	100	262	$4.12\% \pm 7.29\%$
LostAndFound test (NoKnown) [25]	0.12%	15.31%	13 (12)	9 (7)	1203 (1043)	1864 (1709)	$0.08\% \pm 0.16\%$
LiDAR guided Small Obstacle test [26]	0.07%	36.09%	2	6	491	1203	$0.03\% \pm 0.07\%$
Ours: RoadObstacle21 test	0.12%	39.08%	8	31	327	388	$0.10\% \pm 0.25\%$
Dataset (as above)	labels are private	weather conditions	geography				
Fishscapes val	(✓ in test set)	clear	DE				
CAOS BDD test	X	clear, snow, night, rain	US				
Ours: RA21 test	✓	clear, snow	global				
LaF test	X	clear	DE				
Small Obs. test	X	clear	IN				
Ours: RO21 test	✓	clear, snow, night	CH, DE				

Table 1: Main properties of comparable real-world anomaly (top three rows) and obstacle (bottom three rows) segmentation datasets. Our main contribution is the diversity of the anomaly (or obstacle) categories and of the scenes. Note that “void” pixels are not included in this table.

For semantic segmentation, a similar task was therefore proposed by the WildDash benchmark [24] that analyzes semantic segmentation methods trained for driving scenes on a range of failure sources, including full-image anomalies, such as images from the beach. In our work, by contrast, we focus on the problem of robustness to anomalies that only cover a small portion of the image, and on the methods that aim to segment such anomalies, *i.e.* method for the task of *anomaly segmentation*.

One prominent dataset tackling the task of anomaly segmentation is LostAndFound [25], which shares the same setup as Cityscapes [1] but includes anomalous objects / obstacles in various street scenes in Germany. LostAndFound contains 9 different object types as anomalies, and only has annotations for the anomaly and the road surface. Furthermore, it considers children and bicycles as anomalies, even though they are part of the Cityscapes training set, and it contains several labeling mistakes. Although we filter and refine LostAndFound in this work³, similar to Fishscapes [7], the low diversity of anomalies persists.

The CAOS BDD-Anomaly benchmark [8] suffers from a similar low-diversity issue, arising from its use of only 3 object classes sourced from the BDD100k dataset [2] as anomalies (besides including several labeling mistakes, see Appendix F.5). Both Fishscapes and CAOS try to mitigate this low diversity by complementing their real images with synthetic data. Such synthetic data, however, is not realistic and not representative of the situations that can arise in the real world.

In general, the above works illustrate the shortage of diverse real-world data for anomaly segmentation. Additional efforts in this regard have been made by sourcing and annotating images of animals in street scenes [14], and by leveraging multiple sensors, including mainly LiDAR, to detect obstacles on the road [26]. In any event, most of the above datasets are fully published with annotations, allowing methods to overfit to the available anomalies. Furthermore, apart from Fishscapes, we did not find any public leader boards that allow for a trustworthy comparison of new methods. To provide a more reliable test setup, we do not share the labels and request predictions of the shared images to be submitted to our servers. Furthermore, we provide a leader board, which we publish alongside two novel real-world datasets, namely RoadAnomaly21 and RoadObstacle21. A summary of the main properties of the mentioned datasets is given in Table 1. Our main contribution in both proposed datasets is the diversity of the anomaly categories and of the scenes.

In RoadAnomaly21, anomalies can appear anywhere in the image, which is comparable to Fishscapes LostAndFound [7] and CAOS BDD-Anomaly [8]. Although the latter two datasets are larger, their images only show a limited diversity of anomaly types and scenes because they are usually frames of videos captured in single scenes. By contrast, in our dataset every image shows a unique scene, with at least one out of 26 different types of anomalous objects and each sample widely differs in size, ranging from 0.5% to 40% of the image.

In RoadObstacle21, all anomalies (or obstacles) appear on the road, making this dataset comparable to LostAndFound [25] and the LiDAR guided Small Obstacle dataset [26]. Again, the latter two datasets contain more images than ours, however, the high numbers of images result from densely sampling frames from videos. Consequently, those two datasets lack in object diversity (9 and 6 categories, respectively, compared to 31 in our dataset). Furthermore, the videos are recorded under perfect weather conditions, while RoadObstacle21 shows scenes in diverse situations, including night, dirty roads and snowy conditions.

³In the following, we refer to the LostAndFound subset without the images of children, bicycles and invalid annotations as “LostAndFound-NoKnown”.

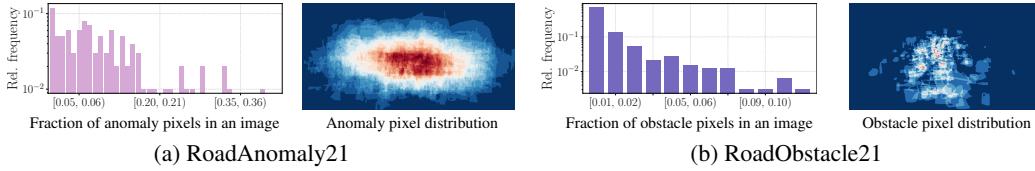


Figure 2: Relative frequency of annotated anomaly / obstacle pixels within an image over the 100 images in the RoadAnomaly21 test dataset (left) and the 327 images in the RoadObstacle21 test dataset (right), respectively. Here, the fraction of anomaly / obstacle pixels serves as a proxy for the size of the objects of interest within an image. Note that the y-axes of the histograms are log scaled.

2.2 Anomaly and Obstacle Segmentation

Anomaly detection was initially tackled in the context of image classification, by developing post-processing techniques aiming to adjust the confidence values produced by a classification DNN [6, 18, 21, 22, 23]. Although originally designed for image-level anomaly detection, most of these methods can easily be adapted to anomaly segmentation [7, 9] by treating each individual pixel in an image as a potential anomaly.

Another relevant approach consists of estimating the uncertainty of the predictions, leveraging the intuition that anomalous image regions should correlate with high uncertainty. One way of doing so is Bayesian (deep) learning [27, 28], where the model parameters are treated as distributions. Because of the computational complexity, approximations to Bayesian inference have been developed [29, 30, 31, 32] and extended to semantic segmentation [33, 34, 35]. Instead of reasoning about uncertainty, other non-Bayesian approaches tune previously-trained models to the task of anomaly detection by either modifying its architecture or exploiting additional data. For example, in [36], anomaly scores are learned by adding a separate branch to the DNN. Instead of modifying the DNNs’s architecture, other approaches [18, 37] incorporate an auxiliary “out-of-distribution” (OoD) dataset during training, which is disjoint from the actual training dataset. These ideas have been employed for anomaly segmentation in [11, 13, 38].

A recent line of work performs anomaly segmentation via generative models that reconstruct / resynthesize the original input image. The intuition is that the reconstructed images will better preserve the appearance of regions containing known objects than those with unknown ones. Pixel-wise anomaly detection is then performed by identifying the discrepancies between the original and reconstructed image. This approach has been used not only for anomaly segmentation [14, 39, 40] but also specifically for road obstacle detection [41, 42, 43].

It is important to note that there are some related works with different definitions of anomaly segmentation. For example, [44] evaluates the segmentation of industrial production anomalies like scratches, and in medical contexts anomaly segmentation can be understood as the detection of diseased parts on e.g. tomography images [45] or brain MRIs [46]. What we define as anomaly segmentation will be discussed in detail in the next Section 3.

3 Benchmark Description

The aim of our benchmark is two-fold. On one hand, by providing diverse data with consistent annotations, we seek to facilitate progress in general semantic anomaly segmentation research. On the other hand, by focusing on road scenes, we expect our benchmark to accelerate the progress towards much needed segmentation/obstacle-detection methods for safe automated driving.

To achieve these goals, our benchmark covers two tasks. First, it tackles the general problem of anomaly segmentation, aiming to identify the image regions containing object classes that have never been seen during training, and thus for which semantic segmentation cannot be correct. This is necessary for any reliable decision making process and it is of great importance to many computer vision applications. Note that, in accordance to [7, 8], we define anomaly as objects that do not fit any of the class definitions in the training data. In some works, anomaly may be used to describe visually different inputs like *e.g.* a car in a novel color, which does not fit our definition.

This strict definition of semantic anomalies, however, can sometimes be ill-defined because (i) existing semantic segmentation datasets, such as Cityscapes [1], often contain ambiguous and ignored regions (annotated as *void*), which are not strictly anomalies since they are seen during training; (ii) the boundary of some classes is fuzzy, *e.g.*, cars *vs.* vans *vs.* rickshaws, making it unclear whether some regions should be considered as anomalous or not. To address these issues, and to account for the fact that automated driving systems need to make sure that the road ahead is free of *any* hazardous objects, we further incorporate obstacle segmentation as a second task in our benchmark, whose goal is to identify any non-drivable region on the road, may the non-drivable region correspond to a known object class or an unknown one.

3.1 Benchmark Tracks and Datasets

We now present the two tracks in our benchmark, corresponding to the two tasks discussed above. Each track contains its own dataset with different properties and is therefore evaluated separately in our benchmark suite. An overview comparing our datasets to related public ones is given in [Table 1](#).

RoadAnomaly21. The road anomaly track benchmarks general anomaly segmentation in full street scenes. It consists of an evaluation dataset of 100 images with pixel-level annotations. The data is an extension of the one introduced in [14], now including a broader collection of images and finer-grain labeling. In particular, we removed low quality images and ones lacking clear road scenes. Besides, we removed labeling mistakes, added the void class and included 68 newly collected images. Each image contains at least one anomalous object, *e.g.*, an animal or an unknown vehicle. The anomalies can appear anywhere in the image, which were collected from web resources and therefore depict a wide variety of environments. The distribution of object sizes and location is shown in [Figure 2\(a\)](#). Moreover, we provide 10 additional images with annotations such that users can check the compatibility of their methods with our benchmark implementation.

RoadObstacle21. The road obstacle track focuses on safety for automated driving. The objects to segment in the evaluation data always appear on the road ahead, *i.e.* they represent realistic and hazardous obstacles that are critical to detect. Our dataset consists of 222 new images taken by ourselves and 105 from [42], summing up to a total of 327 evaluation images with pixel-level annotations. The region of interest in these images is given by the road, which is assumed to belong to the known classes on which the algorithm was trained. The obstacles in this dataset are chosen such that they all can be understood as anomalous objects as well, *e.g.*, stuffed toys, sleighs or tree stumps. They appear at different distances (one distance per image) and are surrounded by road pixels. This allows us to focus our evaluation on the obstacles, as other objects lie outside the region of interest. The distribution of object sizes and location is shown in [Figure 2\(b\)](#). Moreover, this dataset incorporates different road surfaces, lighting and weather conditions, thus encompassing a broad diversity of scenes. An extra track of additional 85 images with scenes at night and in extreme weather, such as snowstorms, is also available. However, the latter subset is excluded from our numerical experiments due to the significant domain shift. Lastly, we provide 30 additional images with annotations such that users can check the compatibility of their methods with our benchmark implementation.

Labeling Policy. In both datasets, the pixel-level annotations include three classes: **1)** anomaly / obstacle, **2)** not anomaly / not obstacle, and **3)** void.

In RoadAnomaly21, the 19 Cityscapes evaluation classes [1], on which most semantic segmentation DNNs are trained, serve as basis to judge whether an object is considered anomalous or not. Everything that fits in the class definitions of Cityscapes is thus labeled as *not anomaly*. This track focuses on the detection of objects which are semantically different from those in the Cityscapes training data. Therefore, if image regions cannot be clearly assigned to any of the Cityscapes classes, they are labeled as *anomaly*. The objects, which are not the main anomalies of interest in the context of street scenes, are labeled as *void* and excluded from our evaluation. The latter class include, for instance, mountains or water in the image background, and street lights. In ambiguous cases, which *e.g.* can arise from a strong domain shift to Cityscapes, we assign the void class as well to properly evaluate semantic anomaly segmentation.

In RoadObstacle21, the task is defined as distinguishing between drivable area and non-drivable area. The goal is to make sure that the road ahead of the ego-car is free of any hazard, irrespective of the object category of potential obstacles. Therefore, the drivable area is labeled as *not obstacle*. This class particularly also includes regions on the road, which visually differ from the rest of the road.



Figure 3: Illustration of the ordinary component-wise intersection over union (IoU) and the adjusted one (sIoU). In both examples above, the prediction \hat{k} (blue rectangle) is the same but covers different targets (green rectangles). On the left, both IoU and sIoU yield the same score. On the right, IoU punishes the prediction as it does not cover each object precisely. By contrast, sIoU checks how much the predictions cover the ground-truth regions, independently of whether prediction/ground truth belongs to a single or multiple objects. In automated driving, it is more important to detect all anomalous regions (whether they belong to single or multiple objects), rather than to detect each object precisely. Since two targets are separated by at least one pixel, $\text{IoU} = \text{sIoU} = 1$ if and only if the prediction covers one target perfectly.

Moreover, every object, which is visually enclosed in the drivable area, is labeled as *obstacle*. All image regions outside the road are assigned to the *void* class and ignored in the evaluation.

As a quality assessment for both tracks, each labeled image was reviewed by at least three people in order to guarantee the highest quality of labels.

3.2 Performance Metrics

For the sake of brevity, in what follows we refer to both anomalies and obstacles as *anomalies*.

Pixel level. Let \mathcal{Z} denote the set of image pixel locations. A model with a binary classifier providing anomaly scores $s(x) \in \mathbb{R}^{|\mathcal{Z}|}$ for an image $x \in \mathcal{X}$ (from a dataset $\mathcal{X} \subseteq [0, 1]^{N \times |\mathcal{Z}| \times 3}$ of N images) discriminates between the two classes *anomaly* and *non-anomaly*. We evaluate the separability of the pixel-wise anomaly scores via the area under the precision-recall curve (AuPRC), where precision and recall are considered as functions of some threshold $\delta \in \mathbb{R}$ applied to $s(x) \forall x \in \mathcal{X}$. The AuPRC puts emphasis on detecting the minority class, making it particularly well suited as our main pixel-wise evaluation metric since the pixel-wise class distributions of RoadAnomaly21 and RoadObstacle21 are considerably unbalanced, *c.f.* Table 1.

To consider the safety point of view, we also include the false positive rate at 95% true positive rate (FPR_{95}) in our evaluation. The FPR_{95} metric indicates how many false positive predictions must be made to reach the desired true positive rate. Note that, any prediction which is contained in a ground-truth labeled region of the class *void* is not counted as false positive, *c.f.* Section 3.1. In particular for the RoadObstacle21 dataset the evaluation is therefore restricted to the road area.

Component level. From a practitioner’s perspective, it is very important to detect all anomalous regions in the scene, regardless of their size, *i.e.*, the number of pixels they cover. However, pixel-level metrics may neglect small anomalies. While one could thus focus on object detection metrics, the notion of individual objects is in fact not relevant for anomaly (region) detection. To satisfy these requirements, we therefore consider performance metrics acting at the component level.

The main metrics for component-wise evaluation are the numbers of *true-positives* (TP), *false-negatives* (FN) and *false-positives* (FP). Considering anomalies as the positive class, we use a component-wise localization and classification quality measure to define the TP, FN and FP components. Specifically, we define this measure as an adjusted version of the component-wise intersection over union (sIoU), introduced in [47]. In particular, while in [47] the sIoU is computed for predicted components, we consider the sIoU for ground-truth components to compute TP and FN. To compute FP, we employ the positive predictive value (PPV, or component-wise precision) for predicted components as quality measure. We discuss the definitions of these quantities in more detail below.

Let \mathcal{Z}_c be the set of pixel locations labeled with class c = “anomaly” in the dataset \mathcal{X} . We consider a connected component of pixels (where the 8 pixels surrounding pixel z in image $x \in \mathcal{X}$ are taken to be its neighbors) that share the same class label as a *component*. Then, let us denote by $\mathcal{K} \subseteq \mathcal{P}(\mathcal{Z}_c)$, with $\mathcal{P}(\mathcal{S})$ the power set of a set \mathcal{S} , the set of anomaly components according to the ground truth, and by $\hat{\mathcal{K}} \subseteq \mathcal{P}(\mathcal{Z}_c)$ the set of components predicted to be anomalous by some machine learning model.

Formally, the sIoU is a mapping $\text{sIoU} : \mathcal{K} \rightarrow [0, 1]$. For $k \in \mathcal{K}$, it is defined as

$$\text{sIoU}(k) := \frac{|k \cap \hat{K}(k)|}{|(k \cup \hat{K}(k)) \setminus \mathcal{A}(k)|} \quad \text{with} \quad \hat{K}(k) = \bigcup_{\hat{k} \in \hat{\mathcal{K}}, \hat{k} \cap k \neq \emptyset} \hat{k} \quad (1)$$

and $\mathcal{A}(k) = \{z \in k' : k' \in \mathcal{K} \setminus \{k\}\}$. With the adjustment $\mathcal{A}(k)$, the pixels are excluded from the union if and only if they correctly intersect with another ground-truth component $k' \in \mathcal{K}(x)$, which is not equal to k . This may happen when one predicted component covers multiple ground-truth components, as illustrated in [Figure 3](#). Given some threshold $\tau \in [0, 1]$, we then call a target $k \in \mathcal{K}$ TP if $\text{sIoU}(k) > \tau$, and FN otherwise. We refer to [Appendix C.2](#) for qualitative examples of the difference between IoU and sIoU.

For the other error type, *i.e.*, FP, we compute the PPV (or precision) for $\hat{k} \in \hat{\mathcal{K}}$, which is defined as

$$\text{PPV}(\hat{k}) := \frac{|\hat{k} \cap K(\hat{k})|}{|\hat{k}|}, \quad (2)$$

We then call a predicted component $\hat{k} \in \hat{\mathcal{K}}$ FP if $\text{PPV}(\hat{k}) \leq \tau$.

As an overall metric, we additionally include the component-wise F_1 -score defined as

$$F_1(\tau) := \frac{2 \cdot \text{TP}(\tau)}{2 \cdot \text{TP}(\tau) + \text{FN}(\tau) + \text{FP}(\tau)} \in [0, 1], \quad (3)$$

which summarizes the TP, FN and FP quantities (that depend on τ). The component-level metrics allow one to evaluate localization of objects irrespective of their size and hence big objects will not dominate these metrics. In addition, while object detection metrics punish predictions that cover multiple ground-truth objects or vice-versa, our component-level metric does not do so, *c.f.* [Figure 3](#).

3.3 Evaluated Methods

Several anomaly segmentation methods have already been evaluated on our benchmark and constitute our initial leader board. We evaluate at least one method per type discussed in [Section 2.2](#), namely

- *Methods originating from image classification:* **maximum softmax probability** [23], **ODIN** [22], **Mahalanobis distance** [21];
- *Bayesian model uncertainty:* **Monte Carlo (MC) dropout** [35], **ensemble** [32];
- *Learning to identify anomalies:* **learned embedding density** [7], **void classifier** [7], **maximized softmax entropy** [11];
- *Reconstruction via generative models:* **image resynthesis** [14], **SynBoost** [39] and **road inpainting** (obstacle track only) [42].

All methods have an underlying semantic segmentation DNN trained on Cityscapes and provide pixel-wise anomaly scores. A semantic segmentation DNN trained on Cityscapes is also our recommendation as underlying model, however, we leave it up to the participants which network and training data they use. Furthermore, some evaluated methods additionally employ out-of-distribution (OoD) data to tune the anomaly detector. For our set of methods, this would be any data with labels semantically different from the Cityscapes train classes. OoD data is also allowed to be used to alleviate the effects of a potential domain shift. For additional details on the methods, we refer the reader to [Appendix D](#).

4 Numerical Experiments

In our benchmark suite we integrate a default method to generate the anomaly segmentation from pixel-wise anomaly scores. We choose the threshold δ^* , at which one pixel is classified as anomaly, by means of the optimal pixel-wise F_1 -score, that we denote with F_1^* . Then, δ^* is computed as

$$\delta^* = \arg \max_{\delta \in \mathbb{R}} 2 \cdot \text{precision}(\delta) \cdot \text{recall}(\delta) / (\text{precision}(\delta) + \text{recall}(\delta)), \quad (4)$$

subject to $\text{precision}(\delta) + \text{recall}(\delta) > 0 \forall \delta$. In [Appendix E](#) we provide a study where δ^* is varied.

Method	requires OoD data	Pixel-level				Component-level										
		Anomaly scores				$k \in \mathcal{K}$		$\hat{k} \in \hat{\mathcal{K}}$		$\tau = 0.25$		$\tau = 0.50$		$\tau = 0.75$		
		AuPRC ↑	FPR ₉₅ ↓	$F_1^* \uparrow$	sIoU ↑	PPV ↑	FN ↓	FP ↓	$F_1 \uparrow$	FN ↓	FP ↓	$F_1 \uparrow$	FN ↓	FP ↓	$F_1 \uparrow$	$\bar{F}_1 \uparrow$
Maximum softmax [23]	✗	28.0	72.0	34.2	15.5	15.3	204	681	11.6	233	714	5.8	256	744	1.2	5.9
ODIN [22]	✗	33.1	71.7	39.1	19.6	17.9	181	924	12.8	226	985	5.6	254	1043	1.2	6.0
Mahalanobis [21]	✗	30.1	87.0	31.9	14.8	10.2	206	1433	6.4	241	1478	2.4	257	1512	0.6	2.9
MC dropout [35]	✗	28.9	69.5	39.0	20.5	17.3	175	1320	10.4	225	1391	4.4	252	1459	1.2	4.9
Ensemble [32]	✗	17.7	91.1	27.8	16.4	20.8	197	1454	7.3	233	1511	3.2	254	1553	0.9	3.4
Void classifier [7]	✓	36.8	63.5	44.3	21.1	22.1	181	797	14.2	219	845	7.5	253	879	1.6	7.6
Embedding density [7]	✗	37.5	70.8	48.7	33.8	20.5	107	1437	16.7	176	1485	9.4	250	1592	1.3	9.2
Image resynthesis [14]	✗	52.3	25.9	60.5	39.5	11.0	95	1187	20.7	153	1225	13.7	230	1294	4.0	12.9
SynBoost [39]	✓	56.4	61.9	58.0	35.0	18.3	109	1062	20.7	178	1114	11.5	247	1216	2.0	11.5
Maximized entropy [11]	✓	85.5	15.0	77.4	49.2	39.5	85	413	41.5	115	421	35.4	163	439	24.8	34.5

Table 2: Benchmark results for our RoadAnomaly21 dataset. This dataset contains 262 ground-truth components in total. The main performance metrics are highlighted with gray columns.

Method	requires OoD data	Pixel-level				Component-level										
		Anomaly (obstacle) scores				$k \in \mathcal{K}$		$\hat{k} \in \hat{\mathcal{K}}$		$\tau = 0.25$		$\tau = 0.50$		$\tau = 0.75$		
		AuPRC ↑	FPR ₉₅ ↓	$F_1^* \uparrow$	sIoU ↑	PPV ↑	FN ↓	FP ↓	$F_1 \uparrow$	FN ↓	FP ↓	$F_1 \uparrow$	FN ↓	FP ↓	$F_1 \uparrow$	$\bar{F}_1 \uparrow$
Maximum softmax [23]	✗	15.7	16.6	22.5	19.7	15.9	255	1494	13.2	326	1503	6.3	372	1517	1.7	6.9
ODIN [22]	✗	21.2	15.4	29.2	20.7	18.5	260	1072	16.1	312	1079	9.9	362	1093	3.5	10.0
Mahalanobis [21]	✗	20.9	13.1	25.8	14.0	21.8	293	1101	12.0	352	1104	4.7	385	1116	0.4	5.5
MC dropout [35]	✗	3.7	50.6	8.0	6.3	5.8	351	2782	2.3	375	2784	0.8	386	2790	0.1	1.0
Ensemble [32]	✗	1.1	77.2	3.1	8.6	4.7	335	3758	2.5	365	3768	1.1	382	3782	0.3	1.3
Void classifier [7]	✓	9.2	41.5	23.4	6.3	20.3	350	350	9.8	365	350	6.0	381	353	1.9	5.9
Embedding density [7]	✗	0.8	46.4	2.0	35.6	2.9	145	10972	4.2	244	11037	2.5	370	11191	0.3	2.4
Image resynthesis [14]	✗	37.2	4.7	38.8	16.6	20.5	286	743	16.5	334	773	8.9	374	824	2.3	9.5
Road inpainting [42]	✗	52.6	47.1	67.5	57.6	39.5	79	580	48.4	131	586	41.8	240	611	25.8	40.2
SynBoost [39]	✓	70.3	3.1	70.1	44.3	41.8	133	352	51.3	185	363	42.6	286	414	22.6	40.4
Maximized entropy [11]	✓	85.1	0.8	79.6	47.9	62.6	136	151	63.7	177	158	55.7	247	174	40.1	54.2

Table 3: Benchmark results for our RoadObstacle21 dataset. This dataset contains 388 ground-truth components in total. The main performance metrics are highlighted with gray columns.

Moreover, for the anomaly track, components smaller than 500 pixels are discarded from the predicted segmentation, and for the obstacle track, components smaller than 50 pixels are discarded. These sizes are chosen based on the smallest ground-truth components. All methods presented in [Section 3.3](#) produce anomaly scores for which we apply the default segmentation method. We emphasize that using our proposed default method for anomaly segmentation masks is completely optional. We provide results without filtering by predicted component sizes in [Appendix E](#). We allow and encourage competitors in the benchmark to submit their own anomaly segmentation masks generated via more sophisticated image operations.

In our evaluation, we additionally include the average sIoU per component \bar{sIoU} , which can be computed by averaging sIoU over all ground-truth components $k \in \mathcal{K}$. Analogously, we also include the average PPV per component \bar{PPV} for all predicted components $\hat{k} \in \hat{\mathcal{K}}$. As the number of component-wise TP, FN and FP depends on some threshold τ for sIoU and PPV, respectively (see [Section 3.2](#)), we average these quantities over different thresholds $\tau \in \mathcal{T} = \{0.25, 0.30, \dots, 0.75\}$, similarly to [4], yielding the averaged component-wise F_1 score $\bar{F}_1 = \frac{1}{|\mathcal{T}|} \sum_{\tau \in \mathcal{T}} F_1(\tau)$.

Discussion of the Results. Our benchmark results for RoadAnomaly21 and RoadObstacle21 are summarized in [Table 2](#) and [Table 3](#), respectively. In general, we observe that methods originally designed for image classification, including maximum softmax, ODIN, and Mahalanobis, do not generalize well to anomaly and obstacle segmentation. For methods based on statistics of the Cityscapes dataset, such as Mahalanobis as well as learned embedding density, anomaly detection is typically degraded by the presence of a domain shift. This results in a poor performance, particularly in RoadObstacle21, where various road surfaces can be observed. Interestingly, learned embedding density, MC dropout and the void classifier yield worse performance than maximum softmax on RoadObstacle21, whereas we observe the opposite on RoadAnomaly21.

The detection methods based on generative models, namely image resynthesis and SynBoost, appear to be better suited to both anomaly and obstacle segmentation at pixel as well as component level, clearly being superior to all the approaches discussed previously. This observation also holds for road inpainting in the obstacle track. These autoencoder-based methods are nonetheless limited by their

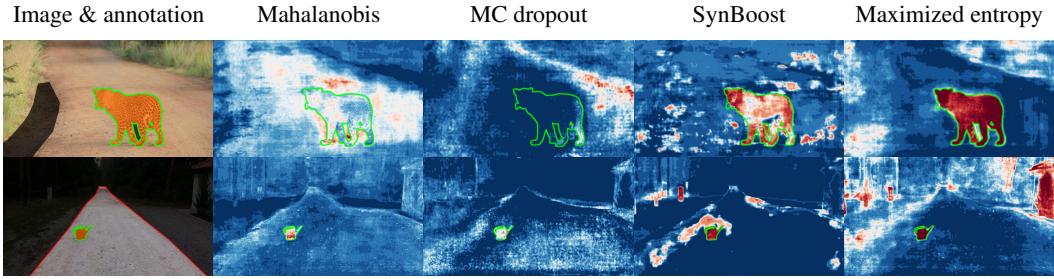


Figure 4: Qualitative comparison of the anomaly scores produced by the methods introduced in Section 3.3 for one example image of RoadAnomaly21 (top row) and one example image of RoadObstacle21 (bottom row). Here, red indicates higher anomaly / obstacle scores and blue lower ones. The ground-truth anomaly / obstacle component is indicated by green contours.

	RoadAnomaly Fishscapes LostAndFound Validation			RoadObstacle			LostAndFound test-NoKnown				
	Pixel-level		Component-level			Pixel-level	Component-level				
Method	OoD data	AuPRC \uparrow $\overline{F_1} \uparrow$	Anomaly scores	$k \in \mathcal{K}$	$\hat{k} \in \hat{\mathcal{K}}$	sIoU \uparrow PPV \uparrow $\overline{F_1} \uparrow$	AuPRC \uparrow $\overline{F_1} \uparrow$	Anomaly scores	$k \in \mathcal{K}$	$\hat{k} \in \hat{\mathcal{K}}$	sIoU \uparrow PPV \uparrow $\overline{F_1} \uparrow$
Maximum softmax [23]	X	28.0	5.9	5.6	40.5	3.5	9.5	1.8	15.7	6.9	30.1
ODIN [22]	X	33.1	6.0	15.5	38.4	9.9	21.9	9.7	21.2	10.0	51.0
Mahalanobis [21]	X	20.0	2.9	32.9	8.7	19.6	29.4	19.2	20.9	5.5	55.0
MC dropout [35]	X	28.9	4.9	14.4	47.8	4.8	18.1	4.3	3.7	1.0	36.2
Ensemble [32]	X	17.7	3.4	0.3	90.4	3.1	1.1	0.4	1.1	1.3	2.9
Void classifier [7]	✓	36.8	7.6	11.7	15.3	9.2	39.1	14.9	9.2	5.9	4.4
Embedding density [7]	X	37.5	9.2	8.9	42.2	5.9	10.8	4.9	0.8	2.4	61.7
Image resynthesis [14]	X	52.3	12.9	5.1	29.8	5.1	12.6	4.1	37.2	9.5	57.1
Road inpainting [42]	X	-	-	-	-	-	-	-	52.6	40.2	83.0
SynBoost [39]	✓	56.4	11.5	64.9	30.9	27.9	48.6	38.0	70.3	40.4	81.8
Maximized entropy [11]	✓	85.5	34.5	44.3	37.7	21.1	48.6	30.0	85.1	54.2	77.9

Table 4: Benchmark results for Fishscapes LostAndFound validation and LostAndFound test-NoKnown, containing 165 and 1709 ground-truth components in total, respectively. In this table the pixel-wise AuPRC and the component-wise $\overline{F_1}$ from RoadAnomaly21 and RoadObstacle21, *c.f.* Table 2 and Table 3, are additionally included for cross evaluation (gray columns).

discrepancy module, and they are outperformed in our experiments by maximized softmax entropy, which peaks at an AuPRC of 86% and a component-wise $\overline{F_1}$ of 49%. This highlights the importance of anomaly and obstacle proxy data. Illustrative example score maps produced by the discussed methods are shown in Figure 4.

In summary, the component-level evaluation highlights the methods’ weaknesses even more clearly than the pixel-wise evaluation, the latter giving a stronger weight to larger anomalies and obstacles. All methods indeed tend to face difficulties in the presence of smaller anomalies and obstacles, as we demonstrate in more detail in Appendix H. In addition, we observe a much lower component-wise $\overline{F_1}$ score than a pixel-wise F_1^* , demonstrating the importance of evaluating at component level. The results w.r.t. the different categories of methods are challenging for models, hence leaving room for improvement.

Our benchmark suite enables a unified evaluation across different datasets whenever ground truth is available. In Table 4 we summarize our results for Fishscapes LostAndFound [7], a validation set of 100 LostAndFound images [25] with refined labels fitting the anomaly track, and the LostAndFound test split, with original labels fitting the obstacle track. Note that, for the LostAndFound test split, we filtered out all images that contain humans and bicycles labeled as obstacles (therefore called LostAndFound test-NoKnown) because we applied anomaly segmentation methods out of the box to the task of obstacle segmentation, and these methods focus on previously-unseen objects.

In comparison to our datasets, for both LostAndFound datasets we observe a less pronounced gap, in terms of both main performance metrics, the pixel-level AuPRC and component-level $\overline{F_1}$ scores, between the methods originally designed for image classification, especially ODIN and Mahalanobis, and those specifically designed for anomaly segmentation, especially road inpainting and maximized entropy. This signals that both of our datasets contribute new challenges for anomaly and obstacle

segmentation. In Appendix H and Appendix I we provide further and more fragmented results in terms of both objects sizes and object categories.

Finally, we also applied our benchmark suite to the LiDAR guided Small obstacle Segmentation dataset [26]. Our main findings are that our whole set of methods yields weak performance on that dataset. The main purpose of this dataset is the detection of small obstacles from multiple sensors including LiDAR. Hence, the conditions for the other sensor modalities are purposely challenging (*e.g.*, low illumination), making this dataset less suitable to camera-only methods. We present the corresponding results in Appendix F.4.

5 Conclusion

In this work, we have introduced a unified and publicly available benchmark suite that evaluates a method’s performance for anomaly segmentation with established pixel level as well as recent component level metrics. Our benchmark suite is applicable in a plug and play fashion to any dataset for anomaly segmentation that comes with ground truth, such as LostAndFound and Fishyscapes LostAndFound, allowing for a better comparison of new methods. Moreover, our benchmark is accompanied with two publicly available datasets, RoadAnomaly21 for anomaly segmentation and RoadObstacle21 for obstacle segmentation.

These two datasets challenge two important abilities of computer vision systems: On one hand their ability to detect and localize unknown objects; on the other hand their ability to reliably detect and localize obstacles on the road, may they be known or unknown. Our datasets consist of real images with pixel-level annotations and depict street scenes with higher variability in object types and object sizes than existing datasets. Our experiments have demonstrated that both of our datasets show a distinct separation in terms of performance between the methods that are specifically designed for anomaly / obstacle segmentation and those that are not. However, there remains much room for performance improvement, particularly in terms of component-wise metrics, which stresses the need for future research in the direction of anomaly segmentation.

The images of the datasets and the software are available at <https://www.segmentmeifyoucan.com/>.

Broader Impact

This benchmark advances research towards the safe deployment of autonomous vehicles. This ultimately will have many consequences, *e.g.*, reducing the number of jobs in the transport sector. More immediately, the benchmark measures the reliability of algorithms and therefore may be misunderstood as giving safety guarantees. This benchmark however only works for the specified training regime *i.e.* it cannot certify fitness for real-world deployment and should not be misunderstood as such. In particular, while our datasets greatly contribute to the diversity of anomalies, the scale of the datasets is still not even close to sufficient in order to represent every possible type of an anomaly. Furthermore, although we do not publicly provide test labels, there remains a risk, common to any other benchmark, of the community designing methods that overfit on our benchmark tasks.

Acknowledgement

Robin Chan and Svenja Uhlemeyer acknowledge funding by the German Federal Ministry for Economic Affairs and Energy, within the projects “KI Absicherung - Safe AI for Automated Driving”, grant no. 19A19005R, and “KI Delta Learning - Scalable AI for Automated Driving”, grant no. 19A19013Q, respectively. We thank the consortiums for the successful cooperation. We would also like to thank the “BUW-KI” team who substantially contributed to collecting and labeling of data.

References

- [1] Marius Cordts, Mohamed Omran, Sebastian Ramos, et al. “The Cityscapes Dataset for Semantic Urban Scene Understanding”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.
- [2] Fisher Yu et al. “Bdd100k: A diverse driving dataset for heterogeneous multitask learning”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 2636–2645.
- [3] Jakob Geyer et al. *A2D2: AEV Autonomous Driving Dataset*. <http://www.a2d2.audi>. 2019.
- [4] Tsung-Yi Lin et al. “Microsoft COCO: Common Objects in Context”. In: *Computer Vision – ECCV 2014*. Ed. by David Fleet et al. Springer International Publishing, 2014, pp. 740–755. ISBN: 978-3-319-10602-1.
- [5] Xiang Zhang and Yann LeCun. “Universum prescription: Regularization using unlabeled data”. In: *Thirty-First AAAI Conference on Artificial Intelligence*. 2017.
- [6] Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. “Why ReLU Networks Yield High-Confidence Predictions Far Away From the Training Data and How to Mitigate the Problem”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.
- [7] Hermann Blum et al. “Fishyscapes: A Benchmark for Safe Semantic Segmentation in Autonomous Driving”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*. Oct. 2019.
- [8] Dan Hendrycks et al. *Scaling Out-of-Distribution Detection for Real-World Settings*. 2020. arXiv: [1911.11132 \[cs.CV\]](https://arxiv.org/abs/1911.11132).
- [9] Matt Angus, Krzysztof Czarnecki, and Rick Salay. “Efficacy of Pixel-Level OOD Detection for Semantic Segmentation”. In: *CoRR* abs/1911.02897 (2019). arXiv: [1911.02897](https://arxiv.org/abs/1911.02897). URL: <http://arxiv.org/abs/1911.02897>.
- [10] Dominik Brüggemann et al. “Detecting Out of Distribution Objects in Semantic Segmentation of Street Scenes”. In: *The 30th European Safety and Reliability Conference (ESREL)*. 2020.
- [11] Robin Chan, Matthias Rottmann, and Hanno Gottschalk. “Entropy Maximization and Meta Classification for Out-Of-Distribution Detection in Semantic Segmentation”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2021, pp. 5128–5137.
- [12] S. Isobe and S. Arai. “Deep convolutional encoder-decoder network with model uncertainty for semantic segmentation”. In: *2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*. 2017, pp. 365–370.
- [13] Nicolas Jourdan, Eike Rehder, and Uwe Franke. “Identification of Uncertainty in Artificial Neural Networks”. In: *Proceedings of the 13th Uni-DAS e.V. Workshop Fahrerassistenz und automatisiertes Fahren*. July 2020.
- [14] Krzysztof Lis et al. “Detecting the Unexpected via Image Resynthesis”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019.
- [15] A. Mehrtash et al. “Confidence Calibration and Predictive Uncertainty Estimation for Deep Medical Image Segmentation”. In: *IEEE Transactions on Medical Imaging* (2020), pp. 1–1.
- [16] Philipp Oberdiek, Matthias Rottmann, and Gernot A. Fink. “Detection and Retrieval of Out-of-Distribution Objects in Semantic Segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2020.
- [17] Hyunsun Choi, Eric Jang, and Alexander A Alemi. *WAIC, but Why? Generative Ensembles for Robust Anomaly Detection*. Oct. 2018. arXiv: [1810.01392 \[stat.ML\]](https://arxiv.org/abs/1810.01392).
- [18] Alexander Meinke and Matthias Hein. “Towards neural networks that provably know when they don’t know”. In: *International Conference on Learning Representations*. 2020. URL: <https://openreview.net/forum?id=ByxGkySKwH>.
- [19] Joost van Amersfoort et al. *Simple and Scalable Epistemic Uncertainty Estimation Using a Single Deep Deterministic Neural Network*. Mar. 2020. arXiv: [2003.02037 \[cs.LG\]](https://arxiv.org/abs/2003.02037).
- [20] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. “Human-level concept learning through probabilistic program induction”. In: *Science* (2015), pp. 1332–1338.

- [21] Kimin Lee et al. “A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks”. In: *Advances in Neural Information Processing Systems*. Ed. by S Bengio et al. Vol. 31. Curran Associates, Inc., 2018, pp. 7167–7177. URL: <https://proceedings.neurips.cc/paper/2018/file/abdeb6f575ac5c6676b747bca8d09cc2-Paper.pdf>.
- [22] Shiyu Liang, Yixuan Li, and R. Srikant. “Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks”. In: *International Conference on Learning Representations*. 2018. URL: <https://openreview.net/forum?id=H1VGkIxRZ>.
- [23] Dan Hendrycks and Kevin Gimpel. “A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks”. In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. 2017. URL: <https://openreview.net/forum?id=Hkg4TI9x1>.
- [24] Oliver Zendel et al. “Wilddash-creating hazard-aware benchmarks”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. openaccess.thecvf.com, 2018, pp. 402–416. DOI: [10.1007/978-3-030-01231-1_25](https://doi.org/10.1007/978-3-030-01231-1_25).
- [25] Peter Pinggera et al. “Lost and found: detecting small road hazards for self-driving vehicles”. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2016.
- [26] Aasheesh Singh et al. *LiDAR guided Small obstacle Segmentation*. Mar. 2020. arXiv: [2003.05970 \[cs.R0\]](https://arxiv.org/abs/2003.05970).
- [27] David J. C. MacKay. “A Practical Bayesian Framework for Backpropagation Networks”. In: *Neural Computation* 4.3 (1992), pp. 448–472.
- [28] Radford M Neal. *Bayesian learning for neural networks*. Vol. 118. Springer Science & Business Media, 2012.
- [29] Yarin Gal and Zoubin Ghahramani. “Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning”. In: *Proceedings of The 33rd International Conference on Machine Learning*. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, June 2016, pp. 1050–1059. URL: <http://proceedings.mlr.press/v48/gal16.html>.
- [30] Andrei Atanov, Arsenii Ashukha, Dmitry Molchanov, et al. “Uncertainty Estimation via Stochastic Batch Normalization”. In: *Advances in Neural Networks – ISNN 2019*. Cham: Springer International Publishing, 2019, pp. 261–269.
- [31] Fredrik K. Gustafsson, Martin Danelljan, and Thomas Bo Schön. “Evaluating Scalable Bayesian Deep Learning Methods for Robust Computer Vision”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2020, pp. 1289–1298.
- [32] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. “Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles”. In: *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 2017, pp. 6402–6413. URL: <http://papers.nips.cc/paper/7219-simple-and-scalable-predictive-uncertainty-estimation-using-deep-ensembles.pdf>.
- [33] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. “Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding”. In: *Proceedings of the British Machine Vision Conference (BMVC)*. BMVA Press, Sept. 2017, pp. 57.1–57.12. ISBN: 1-901725-60-X. DOI: [10.5244/C.31.57](https://doi.org/10.5244/C.31.57). URL: <https://dx.doi.org/10.5244/C.31.57>.
- [34] Alex Kendall and Yarin Gal. “What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?” In: *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 2017, pp. 5574–5584. URL: <http://papers.nips.cc/paper/7141-what-uncertainties-do-we-need-in-bayesian-deep-learning-for-computer-vision.pdf>.
- [35] Jishnu Mukhoti and Yarin Gal. *Evaluating Bayesian Deep Learning Methods for Semantic Segmentation*. 2019. arXiv: [1811.12709 \[cs.CV\]](https://arxiv.org/abs/1811.12709).
- [36] Terrance DeVries and Graham W. Taylor. *Learning Confidence for Out-of-Distribution Detection in Neural Networks*. Feb. 2018. arXiv: [1802.04865](https://arxiv.org/abs/1802.04865). URL: [http://arxiv.org/abs/1802.04865](https://arxiv.org/abs/1802.04865).
- [37] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. “Deep Anomaly Detection with Outlier Exposure”. In: *Proceedings of the International Conference on Learning Representations* (2019).

- [38] Petra Bevandić et al. “Simultaneous Semantic Segmentation and Outlier Detection in Presence of Domain Shift”. In: *Pattern Recognition*. Ed. by Gernot A. Fink, Simone Frintrop, and Xiaoyi Jiang. Cham: Springer International Publishing, 2019, pp. 33–47.
- [39] Giancarlo Di Biase et al. “Pixel-Wise Anomaly Detection in Complex Driving Scenes”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2021, pp. 16918–16927.
- [40] Yingda Xia et al. “Synthesize then Compare: Detecting Failures and Anomalies for Semantic Segmentation”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2020.
- [41] Clement Creusot and Asim Munawar. “Real-time small obstacle detection on highways using compressive RBM road reconstruction”. In: *2015 IEEE Intelligent Vehicles Symposium (IV)*. 2015, pp. 162–167.
- [42] Krzysztof Lis et al. *Detecting Road Obstacles by Erasing Them*. 2020. arXiv: [2012.13633 \[cs.CV\]](https://arxiv.org/abs/2012.13633).
- [43] A. Munawar, P. Vinayavekhin, and G. De Magistris. “Limiting the reconstruction capability of generative neural network using negative learning”. In: *2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*. 2017, pp. 1–6. DOI: [10.1109/MLSP.2017.8168155](https://doi.org/10.1109/MLSP.2017.8168155).
- [44] Paul Bergmann et al. “MVTec AD—A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9592–9600.
- [45] Philipp Seeböck et al. “Exploiting Epistemic Uncertainty of Anatomy Segmentation for Anomaly Detection in Retinal OCT”. In: *IEEE Trans. Medical Imaging* 39.1 (2020), pp. 87–98.
- [46] Christoph Baur et al. “Autoencoders for unsupervised anomaly segmentation in brain MR images: A comparative study”. In: *Medical Image Analysis* 69 (2021), p. 101952. ISSN: 1361-8415. DOI: <https://doi.org/10.1016/j.media.2020.101952>. URL: <https://www.sciencedirect.com/science/article/pii/S1361841520303169>.
- [47] Matthias Rottmann et al. “Prediction Error Meta Classification in Semantic Segmentation: Detection via Aggregated Dispersion Measures of Softmax Probabilities”. In: *2020 IEEE International Joint Conference on Neural Networks (IJCNN)*. 2020.
- [48] Timnit Gebru et al. “Datasheets for Datasets”. In: (Mar. 2018). arXiv: [1803.09010 \[cs.DB\]](https://arxiv.org/abs/1803.09010).
- [49] Kendrick Boyd, Kevin H. Eng, and C. David Page. “Area under the Precision-Recall Curve: Point Estimates and Confidence Intervals”. In: *Machine Learning and Knowledge Discovery in Databases*. Ed. by Hendrik Blockeel et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 451–466.
- [50] L. Chen et al. “DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 40. 2018, pp. 834–848. DOI: [10.1109/TPAMI.2017.2699184](https://doi.org/10.1109/TPAMI.2017.2699184).
- [51] Liang Chieh Chen et al. “Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation”. In: *The European Conference on Computer Vision (ECCV)*. Sept. 2018.
- [52] Yi Zhu, Karan Sapra, Fitsum A. Reda, et al. “Improving Semantic Segmentation via Video Propagation and Label Relaxation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.
- [53] Hengshuang Zhao et al. “Pyramid Scene Parsing Network”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017.

Appendix

A NeurIPS Questionnaire

1. Submission introducing new datasets must include the following in the supplementary materials:
 - (a) Dataset documentation and intended uses. Recommended documentation frameworks include datasheets for datasets, dataset nutrition labels, data statements for NLP, and accountability frameworks.
[Yes] We provide the complete ‘datasheet for datasets’ in Appendix B.
 - (b) URL to website/platform where the dataset/benchmark can be viewed and downloaded by the reviewers.
[Yes] <https://segmentmeifyoucan.com/>
 - (c) Author statement that they bear all responsibility in case of violation of rights, etc., and confirmation of the data license.
[Yes] All authors bear responsibility in case of violation of rights, etc. Confirmation of the data license is given the repository items of <https://zenodo.org/communities/segmentmeifyoucan>.
 - (d) Hosting, licensing, and maintenance plan. The choice of hosting platform is yours, as long as you ensure access to the data (possibly through a curated interface) and will provide the necessary maintenance.
[Yes] To ensure good availability, we chose professionally maintained platforms. Data is hosted at the public data repository zenodo.com and the benchmark website is hosted through github.com. Necessary maintenance such as updating the benchmark record etc. is shared between 3 different research groups such that there is always at least one person reachable.
2. To ensure accessibility, the supplementary materials for datasets must include the following:
 - (a) Links to access the dataset and its metadata. This can be hidden upon submission if the dataset is not yet publicly available but must be added in the camera-ready version. In select cases, e.g when the data can only be released at a later date, this can be added afterward. Simulation environments should link to (open source) code repositories.
[Yes] In general, all data is listed on <https://segmentmeifyoucan.com/>, and metadata more specifically in the zenodo mirrors: <https://zenodo.org/communities/segmentmeifyoucan>
 - (b) The dataset itself should ideally use an open and widely used data format. Provide a detailed explanation on how the dataset can be read. For simulation environments, use existing frameworks or explain how they can be used.
[Yes] The data is stored in standard formats: png, webp, json. We provide ready-to-use code that reads the data at <https://github.com/SegmentMeIfYouCan/road-anomaly-benchmark>.
 - (c) Long-term preservation: It must be clear that the dataset will be available for a long time, either by uploading to a data repository or by explaining how the authors themselves will ensure this.
[Yes] The data is uploaded to multiple mirrors, one of them is the public data repository zenodo.org.
 - (d) Explicit license: Authors must choose a license, ideally a CC license for datasets, or an open source license for code (e.g. RL environments).
[Yes] All images in the obstacle track were recorded by the authors of this work and are published under CC-BY 4.0 license. The images of the anomaly track are all publicly available and licensed as one of {public domain, CC-BY, CC-BY-SA}. A complete list of images, licenses and creators is published as part of the data record: <https://zenodo.org/record/5185336>.
 - (e) Add structured metadata to a dataset’s meta-data page using Web standards (like schema.org and DCAT): This allows it to be discovered and organized by anyone. If you use an existing data repository, this is often done automatically.
[Yes] Metadata is part of the records on zenodo and accessible via different APIs, e.g. https://zenodo.org/oai2d?verb=ListRecords&set=user-segmentmeifyoucan&metadataPrefix=oai_dc.

- (f) Highly recommended: a persistent dereferenceable identifier (e.g. a DOI minted by a data repository or a prefix on identifiers.org) for datasets, or a code repository (e.g. GitHub, GitLab,...) for code. If this is not possible or useful, please explain why.
- anomaly track data <https://doi.org/10.5281/zenodo.5185335>
 - obstacle track data <https://doi.org/10.5281/zenodo.5186546>
 - code repository is on GitHub <https://github.com/SegmentMeIfYouCan/road-anomaly-benchmark>
3. For benchmarks, the supplementary materials must ensure that all results are easily reproducible. Where possible, use a reproducibility framework such as the ML reproducibility checklist, or otherwise guarantee that all results can be easily reproduced, i.e. all necessary datasets, code, and evaluation procedures must be accessible and documented.
[Yes] While, as a public benchmark, we do not give access to the test labels and therefore nobody else should be able to produce the same measurements, we document all code that is used to create the benchmark results (directly yielding the *results.json* that is used for updating the public leaderboard on the website). Further, we created a small validation datasets that allows researchers to check that their method runs as intended. For these validation datasets, we report results in [table 10](#) and [table 11](#) which can be reproduced with the set of methods included in our benchmark suite.
4. For papers introducing best practices in creating or curating datasets and benchmarks, the above supplementary materials are not required. [N/A]

B Datasheet for Datasets

The following section is a complete answer to the datasheet questions from [\[48\]](#).

B.1 Motivation

- **For what purpose was the dataset created?** To evaluate and compare anomaly segmentation methods in driving scenes. Such evaluation enables conclusions on how good methods, usually tested on simpler datasets, are, but also facilitates specific method development for autonomous driving.
- **Who created the dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organization)?** The authors of this work created this dataset to find answers to their research questions. In particular, there was no external party ordering or suggesting the creation of such a benchmark.
- **Who funded the creation of the dataset?** See section Acknowledgements.
- **Any other comments?** No.

B.2 Composition

- **What do the instances that comprise the dataset represent (e.g., documents, photos, people, countries)?** The dataset comprise high resolution images of street scenes with unusual objects, which are all annotated on pixel-level. The objects appearing in the anomaly track data were **not** placed artificially and therefore represent naturally occurring anomalies in a global context. For the obstacle track, the objects were selected and placed by the authors, choosing from available objects that can reasonably appear on a street.
- **How many instances are there in total (of each type, if appropriate)?** 100 images for the anomaly track containing 262 ground truth components (+ 10 images for validation), 327 for the obstacle track containing 388 ground truth components (+ 85 images with hard weather or lightning conditions, + 30 images for validation).
- **Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set?** The set of possibly occurring anomalies in driving scenes is boundless. The instances in this dataset are therefore a subset. For the anomaly track, they are a random sample of openly licensed, available images on the web. Therefore, they have a good geographic coverage. For the obstacle track, all images were taken in

Switzerland and Germany. They have a good coverage over weather and seasons, but are highly biased to European context for both the street background and the selected objects.

- **What data does each instance consist of?** Each data point is an RGB image and a corresponding segmentation map.
- **Is there a label or target associated with each instance?** Yes, our labelling policy is described in Section 3.1.
- **Is any information missing from individual instances?** No.
- **Are relationships between individual instances made explicit (e.g., users' movie ratings, social network links)?** No.
- **Are there recommended data splits (e.g., training, development/validation, testing)?** Yes, our data is supposed to be used for testing only and should not be used for training. We supply a small validation split that enables local testing before submission to the benchmark.
- **Are there any errors, sources of noise, or redundancies in the dataset?** The annotations were created by humans and can therefore contain errors.
- **Is the dataset self-contained, or does it link to or otherwise rely on external resources (e.g., websites, tweets, other datasets)?** The dataset is self-contained.
- **Does the dataset contain data that might be considered confidential (e.g., data that is protected by legal privilege or by doctorpatient confidentiality, data that includes the content of individuals' non-public communications)?** No. All used images are licensed to be shared publicly.
- **Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety?** No.
- **Does the dataset relate to people?** People appear in some images of the Anomaly track.
- **Does the dataset identify any subpopulations (e.g., by age, gender)?** No.
- **Is it possible to identify individuals (i.e., one or more natural persons), either directly or indirectly (i.e., in combination with other data) from the dataset?** It is possible to match faces in the dataset to any other database. However, this applies only to the images of the Anomaly track where all images used were already public, so our dataset did not change that. Regarding the images of Obstacle track identifying individuals is **not** possible.
- **Does the dataset contain data that might be considered sensitive in any way (e.g., data that reveals racial or ethnic origins, sexual orientations, religious beliefs, political opinions or union memberships, or locations; financial or health data; biometric or genetic data; forms of government identification, such as social security numbers; criminal history)?** No.
- **Any other comments?** No.

B.3 Collection Process

- **How was the data associated with each instance acquired?** In the obstacle track, the images were taken by the authors. For the anomaly track, openly licensed images from the web were collected. All images were annotated by humans.
- **What mechanisms or procedures were used to collect the data (e.g., hardware apparatus or sensor, manual human curation, software program, software API)?** Manual human curation.
- **If the dataset is a sample from a larger set, what was the sampling strategy (e.g., deterministic, probabilistic with specific sampling probabilities)?** Images in the anomaly track are a random sample of openly licensed, available images from the web, that show street scenes including at least one anomaly and are of sufficiently high quality. In the obstacle track, some images were extracted from sequences and only images at certain distances (at a rough guess) were included.
- **Who was involved in the data collection process (e.g., students, crowdworkers, contractors) and how were they compensated (e.g., how much were crowdworkers paid)?**

The authors and student/research assistants. Everyone involved in the data generation process was employed at a university at the time of collecting and therefore drew a regular salary.

- **Over what timeframe was the data collected?** The images of the Anomaly Track were collected between August 2019 and August 2021. The images of the Obstacle track were collected between August 2020 and August 2021.
- **Were any ethical review processes conducted (e.g., by an institutional review board)?** No.
- **Does the dataset relate to people?** People appear in some images of the Anomaly track.
- **Did you collect the data from the individuals in question directly, or obtain it via third parties or other sources (e.g., websites)?** We obtained this data via third parties or other sources.
- **Were the individuals in question notified about the data collection?** The collected images were already licensed as public domain or creative commons, i.e., licensed to be shared and used.
- **Did the individuals in question consent to the collection and use of their data?** As we used images from the public domain or licensed a creative commons, we did not ask for consent ourselves.
- **Has an analysis of the potential impact of the dataset and its use on data subjects (e.g., a data protection impact analysis) been conducted?** Not beyond the Broader Impact section.
- **Any other comments?** No.

B.4 Preprocessing/cleaning/labeling

- **Was any preprocessing/cleaning/labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)?** The images for the anomaly track were resized and cropped to two different resolutions (1280×720 and 2048×1024).
- **Was the “raw” data saved in addition to the preprocessed/cleaned/labeled data (e.g., to support unanticipated future uses)?** No.
- **Is the software used to preprocess/clean/label the instances available?** The open-source software ImageMagick was used for resizing the images. As labeling tool, LabelMe was used, which is publicly available (<https://github.com/wkentaro/labelme>).
- **Any other comments?** No.

B.5 Uses

- **Has the dataset been used for any tasks already?** Yes, for this paper.
- **Is there a repository that links to any or all papers or systems that use the dataset?** Yes, the public leaderboard on <https://segmentmeifyoucan.com/leaderboard>.
- **What (other) tasks could the dataset be used for?** No other task, since the labels are hidden.
- **Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses?** No.
- **Are there tasks for which the dataset should not be used?** Certification of fitness for deployment would require at least a larger dataset.
- **Any other comments?** No.

B.6 Distribution

- **Will the dataset be distributed to third parties outside of the entity (e.g., company, institution, organization) on behalf of which the dataset was created?** Yes, the images including the labels for the validation set are public. The labels of the test set however will not be distributed.

- **How will the dataset will be distributed (e.g., tarball on website, API, GitHub)?** Via multiple mirrors as zip archives, all listed on the website <https://segmentmeifyoucan.com/datasets>.
- **When will the dataset be distributed?** Now.
- **Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)?** All parts that are distributed are under public domain or creative commons licenses.
- **Have any third parties imposed IP-based or other restrictions on the data associated with the instances?** No.
- **Do any export controls or other regulatory restrictions apply to the dataset or to individual instances?** No.
- **Any other comments?** No.

B.7 Maintenance

- **Who is supporting/hosting/maintaining the dataset?** The authors of this paper.
- **How can the owner/curator/manager of the dataset be contacted (e.g., email address)?** blumh@ethz.ch,
- **Is there an erratum?** No.
- **Will the dataset be updated (e.g., to correct labeling errors, add new instances, delete instances)?** In case that corrections are necessary, all versions are tracked in the zenodo.com data items.
- **If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances (e.g., were individuals in question told that their data would be retained for a fixed period of time and then deleted)?** No. All images are licensed to be shared.
- **Will older versions of the dataset continue to be supported/hosted/maintained?** In case that there would be multiple versions, only the newest will be maintained.
- **If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so?** Yes, we already have plans to incorporate another body of data into the benchmark. Similarly to the two existing datasets, each set of data is treated as a separate instance, but made comparable by using the same metrics. This can also be observed in this paper and the further comparisons that are listed in the Appendix.
- **Any other comments?** No.

C More Details on Evaluation Metrics

C.1 Pixel level

Let \mathcal{Z} denote the set of image pixel locations. A model with a binary classifier providing scores $s(x) \in \mathbb{R}^{|\mathcal{Z}|}$ for an image $x \in \mathcal{X}$ (from a dataset $\mathcal{X} \subseteq [0, 1]^{N \times |\mathcal{Z}| \times 3}$ of N images) discriminates between the two classes anomaly and non-anomaly. We evaluate the separability of the pixel-wise anomaly scores via the area under the precision-recall curve (AuPRC).

Let $\mathcal{Y} \subseteq \{\text{"anomaly"}, \text{"not anomaly"}\}^{N \times |\mathcal{Z}|}$ be the set of ground truth labels per pixel for \mathcal{X} . Analogously, we denote the predicted labels with $\hat{\mathcal{Y}}(\delta)$, obtained by pixel-wise thresholding on $s(x) \forall x \in \mathcal{X}$ w.r.t. some threshold value $\delta \in \mathbb{R}$. Then, for the anomaly class ($c_1 = \text{"anomaly"}$) we compute

$$\text{precision}(\delta) = \frac{|\mathcal{Y}_{c_1} \cap \hat{\mathcal{Y}}_{c_1}(\delta)|}{|\hat{\mathcal{Y}}_{c_1}(\delta)|}, \quad \text{recall}(\delta) = \frac{|\mathcal{Y}_{c_1} \cap \hat{\mathcal{Y}}_{c_1}(\delta)|}{|\mathcal{Y}_{c_1}|} \quad (5)$$

with \mathcal{Y}_{c_1} and $\hat{\mathcal{Y}}_{c_1}$ representing the ground truth labels and predicted labels, respectively. For the AuPRC, precision and recall are considered as functions of δ . The AuPRC approximates

$\int \text{precision}(\delta) d\text{recall}(\delta)$ and is threshold independent [49]. It also puts emphasis on detecting the minority class, making it particularly well suited as our main evaluation metric since the pixel-wise class distributions of RoadAnomaly21 and RoadObstacle21 are considerably unbalanced, *c.f.* section 3.1.

To consider the safety point of view, we also include the false positive rate at 95% true positive rate (FPR_{95}) in our evaluation, where the true positive rate (TPR) is equal to the recall of the anomaly class. The false positive rate (FPR) is the number of pixels falsely predicted as anomaly over the number of all non-anomaly pixels. Hence, for the anomaly class we compute

$$\text{FPR}_{95} = \frac{|\mathcal{Y}_{c_1}(\delta') \cap \mathcal{Y}_{c_2}|}{|\mathcal{Y}_{c_2}|} \quad \text{s.t. } \text{TPR}(\delta') = 0.95 , \quad (6)$$

where c_2 = “not anomaly”. The metric FPR_{95} indicates how many false positive predictions are necessary to guarantee a desired true positive rate. Note that, any prediction which is contained in a ground truth labeled region of class void is not counted as false positive, *c.f.* section 3.1. In particular for the RoadObstacle21 dataset the evaluation is therefore restricted to the road area.

C.2 Component level - Qualitative examples revealing the difference of IoU and sIoU

If we consider component-level metrics over ground-truth components, it may happen that several components are close together and therefore covered by one predicted component. Although the real error can be small, the IoU punishes both ground-truth components. The same holds the other way around when considering metrics over predicted components, *i.e.* when one ground-truth component is covered by several predicted components. A qualitative example is given in figure 5. A small number of incorrectly predicted pixels may cause a strong decrease in the IoU. The adjusted IoU (sIoU) is less sensitive in such cases. sIoU focuses on correctly covering the regions of obstacles/anomalies in the image rather than finding such regions separately for each instance, as done by IoU. In self-driving it is more important to know the regions of anomaly rather than how many of them exist.

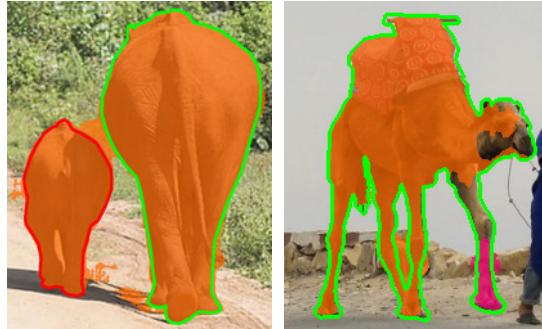


Figure 5: Two examples underlining the difference between IoU and adjusted IoU (sIoU). The ground-truth components are indicated by green/red contours, predicted components are highlighted by other colors. *Left:* Two ground-truth components (green & red) intersect with one predicted component (orange). Green: **IoU 68.18%** vs. **sIoU 87.01%**; red: **IoU 21.68%** vs. **sIoU 68.44%**. *Right:* Two predicted components (orange & pink) intersect with one ground-truth component (green). Orange: **IoU 78.97%** vs. **sIoU 81.69%**; pink: **IoU 0.344%** vs. **sIoU 18.91%**.

D Evaluated Methods

In this section, we first briefly introduce the methods which are evaluated on our benchmark and constitute our initial leader board. Afterwards we additionally provide technical details to those introduced methods.

D.1 Brief Description of Methods

All methods subject to evaluation are stated in boldface in the following. We evaluate at least one method per type discussed in section 2.2. All methods have an underlying semantic segmentation DNN trained on Cityscapes and they all provide pixel-wise anomaly scores.

Given an input image, the **maximum softmax probability** (MSP) of a DNN’s corresponding output is a commonly-used baseline for OoD detection at image level [23]. Adding small perturbations to every pixel of the input image and applying temperature scaling enhances the anomaly detection ability of MSP. The latter approach is known as **ODIN** [22]. Another well-known method detects anomalies based on the **Mahalanobis distance**. It is computed by estimating Gaussian distributions of latent features of a DNN’s penultimate layer, therefore yielding an estimate of the likelihood of a test sample w.r.t. the distribution in the training data. All these methods are originally designed for image classification but can be adapted straightforwardly to segmentation and represent good baselines in our benchmark.

As Bayesian approach to uncertainty estimation we employ **Monte Carlo (MC) dropout** in our evaluation. MC dropout has already been investigated for semantic segmentation. We follow [35] and use the mutual information as pixel-wise anomaly scores, which captures the epistemic uncertainty of a DNN. Furthermore, we additionally evaluate an **ensemble** of semantic segmentation networks.

In [7] several approaches to learning the confidence with respect to the presence of anomalies have been proposed. The **learned embedding density** aims to approximate the distribution of feature embeddings within a DNN via normalizing flows. At test time, the negative log-likelihood for each embedded representation of an image measures the discrepancy of a test embedding with respect to training embeddings, where high discrepancies indicate anomalies. These scores are then upsampled via bilinear interpolation to obtain the pixel-wise anomaly scores. Alternatively, the segmentation DNN can be modified to learn the confidence for the presence of anomalies, requiring an OoD dataset. As in [7], a Cityscapes DNN is trained with an additional model output for the Cityscapes void class. The anomaly scores are then the softmax scores for the that class, therefore this method is called **void classifier**. Additionally, one can also retrain a DNN with a different OoD proxy, such as the COCO dataset [4], and enforce **maximized softmax entropy** [11] on samples of the OoD proxy. All these methods tune previously-trained DNNs to the task of anomaly segmentation and are included in our evaluation.

As autoencoders in our evaluation, we employ **image resynthesis** together with a discrepancy network that extracts meaningful differences based on the information provided by the DNN’s segmentation mask, the resynthesized input image and the original image itself [14]. This approach can be extended by including uncertainty estimates in the discrepancy module, aiming to boost the anomaly segmentation performance, known as **SynBoost** [39]. One method specifically designed for obstacle segmentation is called **road inpainting** [42]. This method inpaints road patches in a sliding window manner. The resulting synthesized image is then again presented to a discrepancy network, similarly as in [14], for pixel-wise obstacle scores.

D.2 Method Description in Detail

All methods provide pixel-wise anomaly scores $s(x) \in \mathbb{R}^{|\mathcal{Z}|}$, $x \in \mathcal{X}$ where anomalies correspond to higher values. As a reminder, \mathcal{Z} denotes the set of image coordinates and $\mathcal{X} \subseteq [0, 1]^{N \times |\mathcal{Z}| \times 3}$ a dataset with N images. Below, we describe how s is obtained for each approach.

Maximum softmax probability. Let $f : \mathcal{X} \rightarrow \mathbb{R}^{|\mathcal{Z}| \times |\mathcal{C}|}$ denote the output of a semantic segmentation DNN. The maximum softmax probability (MSP) is a commonly-used baseline for OoD detection at image level [23]. It computes an anomaly score for each pixel $z \in \mathcal{Z}$ as

$$s_z(x) = 1 - \max_{c \in \mathcal{C}} \sigma(f_z^c(x)), \quad x \in \mathcal{X}, \quad (7)$$

where $\sigma(\cdot) : \mathbb{R}^{|\mathcal{C}|} \rightarrow (0, 1)^{|\mathcal{C}|}$ denotes the softmax function over the non-anomalous class set \mathcal{C} .

ODIN. Let $t \in \mathbb{R} \setminus \{0\}$ be a temperature scaling parameter and $\varepsilon \in \mathbb{R}$ a perturbation magnitude. Following [22] small perturbations are added to every pixel $z \in \mathcal{Z}$ of image x by

$$\tilde{x}_z = x_z - \varepsilon \text{sign} \left(-\frac{\partial}{\partial x_z} \log \max_{c \in \mathcal{C}} \sigma(f_z^c(x)/t) \right). \quad (8)$$

Then, an anomaly score is obtained analogously to equation (7) via the MSP as

$$s_z(x) = 1 - \max_{c \in \mathcal{C}} \sigma(f_z^c(\tilde{x})/t). \quad (9)$$

Mahalanobis distance. Let $h^{L-1}(\cdot)$ denote the output of the penultimate layer of a DNN with $L \in \mathbb{N}$ layers, *i.e.* $f(x) = h^L(x), x \in \mathcal{X}$. Under the assumption that

$$P(h_z^{L-1}(x) | y_z(x) = c) = \mathcal{N}(h_z^{L-1}(x) | \mu^c, \Sigma^c), \quad (10)$$

an anomaly score for each pixel z can be computed as the Mahalanobis distance [21]

$$s_z(x) = \min_{c \in \mathcal{C}} (h_z^{L-1}(x) - \hat{\mu}^c)^T \hat{\Sigma}^{c^{-1}} (h_z^{L-1}(x) - \hat{\mu}^c), \quad (11)$$

where $\hat{\mu}^c$ and $\hat{\Sigma}^c$ are estimates of the class mean μ^c and class covariance Σ^c , respectively, of the latent features in the penultimate layer. This Mahalanobis distance yields an estimate of the likelihood of a test sample with respect to the closest class distribution in the training data, which are assumed to be class-conditional Gaussians.

Monte Carlo dropout. Let $M \in \mathbb{N}$ denote the number of Monte Carlo sampling rounds and $\hat{q}_m^c := \sigma(f_z^c(x))$ the softmax probability of class $c \in \mathcal{C}$ for sample $m \in \{1, \dots, M\}$. The predictive entropy is computed as

$$\hat{E}(f(x)) = - \sum_{c \in \mathcal{C}} \left(\frac{1}{M} \sum_{m=1}^M \hat{q}_m^c \right) \log \left(\frac{1}{M} \sum_{m=1}^M \hat{q}_m^c \right). \quad (12)$$

As suggested in [35], the mutual information can then be used to define an anomaly score

$$s_z(x) = \hat{E}(f(x)) - \frac{1}{M} \sum_{c \in \mathcal{C}} \sum_{m=1}^M \hat{q}_m^c \log(\hat{q}_m^c). \quad (13)$$

Ensemble. Similar to Monte Carlo dropout, multiple samples of softmax probabilities $\hat{q}_m^c := \sigma(f_z^c(x)), c \in \mathcal{C}, m \in \{1, \dots, M\}$ are drawn from multiple semantic segmentation models. Those models have the same network architecture but are trained with different weights initialization [32]. Again, the mutual information is used as anomaly score

$$s_z(x) = \hat{E}(f(x)) - \frac{1}{M} \sum_{c \in \mathcal{C}} \sum_{m=1}^M \hat{q}_m^c \log(\hat{q}_m^c). \quad (14)$$

Void classifier. In [36], an approach to learning the confidence with respect to the presence of anomalies was proposed. Here, we adapt this by using the Cityscapes void class to approximate the anomaly distribution. We then trained a Cityscapes DNN $f : \mathcal{X} \mapsto \mathbb{R}^{|\mathcal{Z}| \times (|\mathcal{C}|+1)}$ with an additional class, *i.e.*, a dustbin [5], and compute the anomaly score for each pixel $z \in \mathcal{Z}$ as the softmax score for the void class, which yields

$$s_z(x) = \sigma(f_z^{\text{void}}(x)), x \in \mathcal{X}. \quad (15)$$

Learned embedding density. Let $h^l(x) \in \mathbb{R}^{|\mathcal{Z}'| \times n_l}, n_l \in \mathbb{N}, \mathcal{Z}' \subset \mathcal{Z}$, be the embedding vector of a segmentation DNN at layer $l \in \{1, \dots, L\}$ for image $x \in \mathcal{X}$. The true distribution $p^*(h^l(x)), x \in \mathcal{X}_{\text{train}} \subset \mathcal{X}$ can be approximated with a normalizing flow $\hat{p}(h^l(x)) \approx p^*(h^l(x))$. At test time, the negative log-likelihood $-\log \hat{p}_{z'}(h^l(x)) \in (0, \infty)$ for each embedding location $z' \in \mathcal{Z}'$ then measures the discrepancy of a test embedding with respect to training embeddings, where higher discrepancies indicate anomalies [7]. The resulting anomaly score map are of size $|\mathcal{Z}'| = \frac{1}{n} |\mathcal{Z}|$, with $n \in \mathbb{N}$ the rescaling factor for \mathcal{Z}' to match the size of \mathcal{Z} , and hence bring back latent features to the full image resolution $|\mathcal{Z}|$ via bilinear interpolation $u : \mathbb{R}^{|\mathcal{Z}'|} \rightarrow \mathbb{R}^{|\mathcal{Z}|}$. This yields an anomaly score for each $z \in \mathcal{Z}$ as

$$s_z(x) = u_z \left((-\log \hat{p}(h_{z'}^l(x)))_{z' \in \mathcal{Z}'} \right), x \in \mathcal{X}. \quad (16)$$

Image resynthesis. The semantic segmentation map $\hat{y}(x) := (\arg \max_{c \in \mathcal{C}} f_z^c(x))_{z \in \mathcal{Z}}$ predicted by a DNN for image $x \in \mathcal{X}$ is passed to a generative network $g : \mathcal{C}^{|\mathcal{Z}|} \rightarrow \mathcal{X}'$ whose goal is to resynthesize x , *i.e.* $x \approx g(\hat{y}(x)) \in \mathcal{X}'$, with \mathcal{X}' the resynthesized input space. Assuming that mislabeled pixels in the segmentation map, *i.e.* anomaly pixels, will be poorly reconstructed, a *discrepancy network* [14] $d : \mathcal{C}^{|\mathcal{Z}|} \times \mathcal{X}' \times \mathcal{X} \rightarrow \mathbb{R}^{|\mathcal{Z}|}$ is trained to extract the meaningful differences

based on the information provided by $\hat{y}(x)$, $g(\hat{y}(x))$ and x itself. The output of $d(\cdot)$ serves as anomaly score for each $z \in \mathcal{Z}$, that is,

$$s_z(x) = d_z(\hat{y}(x), g(\hat{y}(x)), x), \quad x \in \mathcal{X}. \quad (17)$$

Road inpainting. Another approach motivated by image resynthesis is road inpainting, which is specifically designed for obstacle segmentation. This method inpaints patches on the road (that is assumed to be known a-priori) in a sliding window manner and passes the resulting resynthesized image $g'(x)$ to the discrepancy network together with the original input image. Thus, the anomaly score is

$$s_z(x) = d_z(g'(x), x), \quad x \in \mathcal{X}. \quad (18)$$

SynBoost. This approach follows a similar idea as image resynthesis but includes further inputs in the discrepancy module. In particular, for all $z \in \mathcal{Z}$ the pixel-wise softmax entropy

$$H_z(x) = - \sum_{c \in \mathcal{C}} \sigma(f_z^c(x)) \log(\sigma(f_z^c(x))) \quad (19)$$

and the pixel-wise softmax distance

$$D_z(x) = 1 - \max_{c \in \mathcal{C}} \sigma(f_z^c(x)) + \max_{c' \in \mathcal{C} \setminus \{\arg \max_{c \in \mathcal{C}}\}} \sigma(f_z^{c'}(x)) \quad (20)$$

are included. The anomaly score for $x \in \mathcal{X}$ is then obtained via

$$s_z(x) = d_z(\hat{y}(x), g(\hat{y}(x)), x, H(x), D(x)). \quad (21)$$

Maximized entropy. Starting from a pretrained DNN, a second training objective is introduced to maximize the softmax entropy on OoD pixels [11, 13, 37]. This yields the multi-criteria loss function

$$(1 - \lambda) \mathbb{E}_{(x,y) \sim \mathcal{D}_{in}} [\ell_{in}(\sigma(f_z(x)), y_z(x))] + \lambda \mathbb{E}_{x' \sim \mathcal{D}_{out}} [\ell_{out}(\sigma(f_z(x')))], \quad \lambda \in [0, 1], \quad (22)$$

where ℓ_{in} is the empirical cross entropy and ℓ_{out} the averaged negative log-likelihood over all classes for the in-distribution data \mathcal{D}_{in} and the out-distribution data \mathcal{D}_{out} , respectively. To approximate \mathcal{D}_{out} , a subset of the COCO dataset [4] is used whose images do not depict any object classes also available in \mathcal{D}_{in} , which is the Cityscapes dataset [1]. The COCO subset together with the Cityscapes training data are then included into a tender retraining of the pretrained Cityscapes model. The anomaly score is then computed via the softmax entropy as

$$s_z(x) = - \sum_{c \in \mathcal{C}} \sigma(f_z^c(x)) \log(\sigma(f_z^c(x))). \quad (23)$$

D.3 Underlying Segmentation DNNs

Most of our evaluated methods build upon variants of DeepLab [50] network architectures for semantic segmentation. In particular, for MC dropout, void classifier and learned embedding density we use a DeepLabv3+ model with an Xception backbone [51], as presented first in [7]. For maximum softmax, ODIN, Mahalanobis distance and maximized entropy, we employ a more modern DeepLabv3+ model with a WideResNet38 backbone [52]. For image resynthesis we use the more lightweight PSPNet as underlying model for semantic segmentation just like originally proposed by [14]. All these networks are initialized with publicly available weights which are pretrained on the Cityscapes dataset. To show the capacity of the network, we report the mean Intersection over Union (mIoU) on the Cityscapes validation dataset in [Table 5](#).

D.4 Inference Time Comparison

In practice, anomaly segmentation is desired to be obtained in real time. Therefore, we report the run-time of the evaluated anomaly segmentation methods as further performance metric that expresses a method's suitability as online application. We measure the total inference time for RoadAnomaly21, *i.e.* the time from feeding all images through a model to obtaining pixel-wise anomaly scores. Afterwards we average the time per image and report them in [table 5](#). All methods are compared with the same hardware (NVIDIA Quadro P6000), however they might differ in the underlying network architecture.

Method	Semantic segmentation Network architecture	mIoU ↑ on Cityscapes Val.	time in s ↓ per image
Maximum softmax	DeepLabv3+ WideResNet38 backbone [52]	90.3%	1.17
ODIN	DeepLabv3+ WideResNet38 backbone [52]	90.3%	16.74
Mahalanobis Distance	DeepLabv3+ WideResNet38 backbone [52]	90.3%	63.60
MC dropout	DeepLabv3+ Xception backbone [51]	80.3%	19.68
Void Classifier	DeepLabv3+ Xception backbone [51]	80.3%	2.02
Embedding density	DeepLabv3+ Xception backbone [51]	80.3%	10.66
Image resynthesis	PSPNet [53]	79.9%	1.43
SynBoost	DeepLabv3+ WideResNet38 backbone [52]	90.3%	2.09
Maximized entropy	DeepLabv3+ WideResNet38 backbone [52]	89.3%	1.07

Table 5: Run time comparison on a NVIDIA Quadro P6000 for different anomaly segmentation methods. The averaged inference time for one image of RoadAnomaly21 is reported in seconds. Moreover, the mean Intersection over Union (mIoU) on the Cityscapes validation dataset is reported to check whether anomaly segmentation decreases the original semantic segmentation performance.

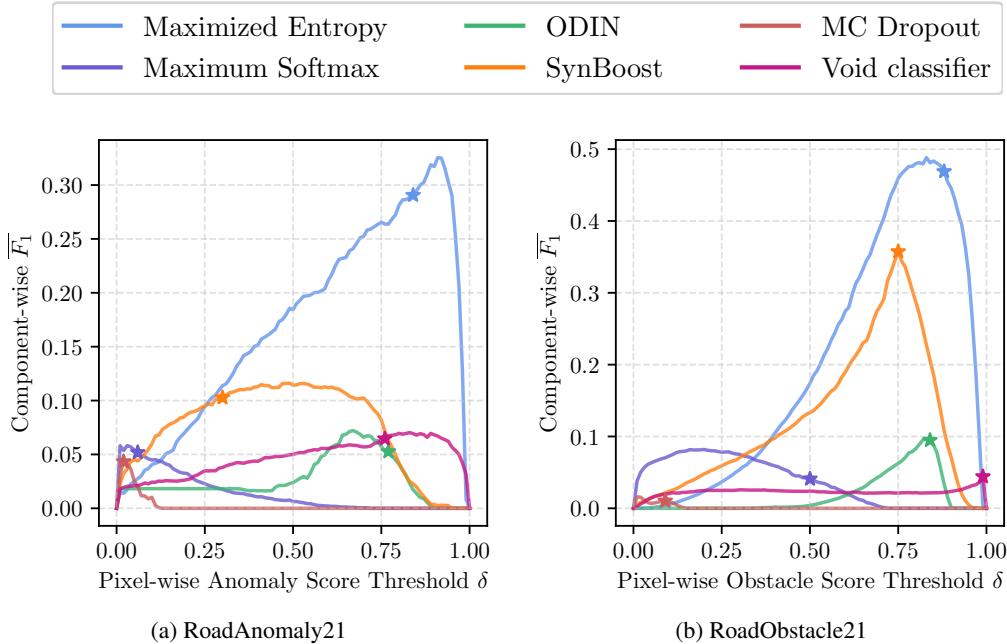


Figure 6: The averaged component-wise metric \bar{F}_1 as function of the pixel-wise anomaly / obstacle threshold δ for RoadAnomaly21 and RoadObstacle21, respectively, *c.f.* table 2 and 3. The “star” marker indicates a method’s \bar{F}_1 -score at the chosen threshold δ^* according to equation (4), which is used in our default procedure for generating segmentation masks from pixel-wise anomaly / obstacle scores. We observe that for most methods δ^* yields near optimal component-wise \bar{F}_1 -scores, however not for every single tested method. Therefore, we encourage competitors to submit their own anomaly segmentation masks based on more sophisticated methods.

E Parameter Study

In our evaluation, the component-wise F_1 score (equation (3)) does not only depend on the parameter τ but also δ . Recall that τ is the threshold for sIoU at which one component is considered to be false negative and true positive, respectively, see also section 3.2. As we generate anomaly segmentation masks from pixel-wise anomaly scores, we introduced another threshold δ at which a given pixel is considered as anomaly. For generating segmentation masks with our default method, we chose that threshold as δ^* (equation (4)) which is the parameter for which a method achieves its best pixel-wise F_1 score, *i.e.* the optimal threshold according to the precision recall curve.

Method	OoD data	Component-level metrics with filtering						Component-level metrics without filtering									
		$k \in \mathcal{K}$			$\hat{k} \in \hat{\mathcal{K}}$			$\tau = 0.50$			$k \in \mathcal{K}$			$\hat{k} \in \hat{\mathcal{K}}$			
		$\text{IoU} \uparrow$	$\text{PPV} \uparrow$	$\text{FN} \downarrow$	$\text{FP} \downarrow$	$F_1 \uparrow$	$\overline{F}_1 \uparrow$	$\text{IoU} \uparrow$	$\text{PPV} \uparrow$	$\text{FN} \downarrow$	$\text{FP} \downarrow$	$F_1 \uparrow$	$\overline{F}_1 \uparrow$	$\text{IoU} \uparrow$	$\text{PPV} \uparrow$	$\text{FN} \downarrow$	
Maximum softmax [23]	X	15.5	15.3	233	714	5.8	5.9	15.4	15.7	232	713	6.0	5.8				
ODIN [22]	X	19.6	17.9	226	985	5.6	6.0	19.7	17.5	227	983	5.5	6.0				
Mahalanobis [21]	X	14.8	10.2	241	1478	2.4	2.9	14.8	10.5	241	1464	2.4	2.9				
MC dropout [35]	X	20.5	17.3	225	1391	4.4	4.9	20.5	17.3	225	1391	4.4	4.9				
Ensemble [32]	X	16.4	20.8	233	1511	3.2	3.4	19.8	12.6	225	1528861	0.0	0.0				
Void classifier [7]	✓	21.1	22.1	219	845	7.5	7.6	21.1	22.1	219	845	7.5	7.6				
Embedding density [7]	X	33.8	20.5	176	1485	9.4	9.2	34.0	20.8	176	1491	9.4	9.2				
Image resynthesis [14]	X	39.5	11.0	153	1225	13.7	12.9	39.6	11.1	152	1225	13.8	13.0				
SynBoost [39]	✓	35.0	18.3	178	1114	11.5	11.5	34.7	17.8	179	1129	11.3	11.2				
Maximized entropy [11]	✓	49.2	39.5	115	421	35.4	34.5	49.2	39.4	115	421	35.4	34.4				

Table 6: Comparison of benchmark results for our RoadAnomaly21 dataset when not using the filtering included in our default segmentation post-processing step. This dataset contains 262 ground-truth components in total. The main performance metrics are highlighted with gray columns.

In this section, we perform a parameter study to show what impact the choice of δ has on the component-wise performance. By considering \overline{F}_1 as component-wise performance metric we already cover varying values for τ , since \overline{F}_1 is the average of component-wise F_1 -scores over different values of τ . The dependence of \overline{F}_1 on the parameter δ is illustrated in figure 6 for RoadAnomaly21 and RoadObstacle21, respectively. For the sake of clarity, we only include six methods in total in this study, with at least one per type as discussed in section 2.2.

We observe that for most of the evaluated methods the choice of δ^* leads to an \overline{F}_1 -score close its optimum, with some methods even reaching their optimal scores at δ^* , e.g. MC dropout on RoadAnomaly21 and SynBoost as well as the void classifier on RoadObstacle21. For the other methods the gap to the optimal \overline{F}_1 -score reaches up to 2.8 percent points for maximized entropy on RoadAnomaly21 and even 4.1 percent points for maximum softmax on RoadObstacle21. However, except for the latter case where the distance between δ^* and the actual optimal location for \overline{F}_1 is 0.30, for all other methods the distance (in terms of \overline{F}_1) of δ^* to the optimal δ is at most 0.05.

This parameter study shows that our default method for generating segmentation masks from pixel-wise anomaly scores via the threshold δ^* is a legitimate choice, reaching a near optimal component-wise performance. Nonetheless, the parameter study also demonstrates that for some methods the \overline{F}_1 -score can still be improved. Consequently, we allow (and encourage) competitors in the benchmark to submit their own anomaly segmentation masks with more sophisticated image operations and other post-processing techniques.

Another parameter included in the computation of the evaluation metrics is the size of predicted components in segmentation masks when generated from pixel-wise score maps. In our default post-processing method, we remove all components smaller than 500 pixels and 50 pixels in the anomaly and obstacle track, respectively, to reduce the amount of false positive components. To use this kind of filtering is completely optional. However, as can be seen in Table 6, Table 7, Table 8 and Table 9, we recommend using the post-processing option when competitors do not include a more sophisticated method. This is also why, we make our post-processing step transparent in this work since the size parameters are based on knowledge of ground truth components.

F Evaluated Datasets

Besides RoadAnomaly21 and RoadObstacle21 we also performed analogous benchmark evaluations for three additional publicly available datasets: Fishscapes LostAndFound [7], LostAndFound test set [25], and the LiDAR guided Small obstacle Segmentation dataset [26]. For the sake of comparison, we chose the Fishscapes LostAndFound validation set for the anomaly track and the LostAndFound test set as well as the Small Obstacle dataset for the obstacle track.

E.1 RoadAnomaly21 & RoadObstacle21 Validation Dataset

In order to ensure that methods run as intended with our benchmark code, we provide small validation sets (including ground truth annotations) for the anomaly track, called *RoadAnomaly21 validation*, and for the obstacle track, called *RoadObstacle21 validation*.

		Component-level metrics with filtering						Component-level metrics without filtering					
Method	OoD data	$k \in \mathcal{K}, \hat{k} \in \hat{\mathcal{K}}$			$\tau = 0.50$			$k \in \mathcal{K}, \hat{k} \in \hat{\mathcal{K}}$			$\tau = 0.50$		
		sIoU ↑	PPV ↑	FN ↓	FP ↓	F_1 ↑	$\overline{F_1}$ ↑	sIoU ↑	PPV ↑	FN ↓	FP ↓	F_1 ↑	$\overline{F_1}$ ↑
Maximum softmax [23]	X	19.7	15.9	326	1503	6.3	6.9	21.5	8.1	325	9624	1.3	1.4
ODIN [22]	X	20.7	18.5	312	1079	9.9	10.0	22.3	9.6	308	7260	2.1	2.1
Mahalanobis [21]	X	14.0	21.8	352	1104	4.7	5.5	17.0	7.5	348	13630	0.6	0.7
MC dropout [35]	X	6.3	5.8	375	2784	0.8	1.0	7.0	2.9	375	20727	0.1	0.1
Ensemble [32]	X	8.6	4.7	365	3768	1.1	1.3	11.0	1.8	364	369439	0.0	0.0
Void classifier [7]	✓	6.3	20.3	365	350	6.0	5.9	2.8	42.8	384	123	1.6	2.6
Embedding density [7]	X	35.6	2.9	244	11037	2.5	2.4	36.1	1.6	246	33598	0.8	0.8
Image resynthesis [14]	X	16.6	20.5	334	773	8.9	9.5	17.4	15.8	332	7003	1.5	1.6
Road inpainting [42]	X	57.6	39.5	131	586	41.8	40.2	59.7	17.2	127	4789	9.6	9.3
SynBoost [39]	✓	44.3	41.8	185	363	42.6	40.4	45.2	22.6	185	1432	20.1	19.2
Maximized entropy [11]	✓	47.9	62.6	177	158	55.7	54.2	48.7	35.1	177	758	31.1	30.3

Table 7: Comparison of benchmark results for our RoadObstacle21 dataset when not using the filtering included in our default segmentation post-processing step. This dataset contains 388 ground-truth components in total. The main performance metrics are highlighted with gray columns.

		Component-level metrics with filtering						Component-level metrics without filtering					
Method	OoD data	$k \in \mathcal{K}, \hat{k} \in \hat{\mathcal{K}}$			$\tau = 0.50$			$k \in \mathcal{K}, \hat{k} \in \hat{\mathcal{K}}$			$\tau = 0.50$		
		sIoU ↑	PPV ↑	FN ↓	FP ↓	F_1 ↑	$\overline{F_1}$ ↑	sIoU ↑	PPV ↑	FN ↓	FP ↓	F_1 ↑	$\overline{F_1}$ ↑
Maximum softmax [23]	X	3.5	9.5	164	199	0.5	1.8	11.7	3.1	159	23134	0.1	0.1
ODIN [22]	X	9.9	21.9	146	142	11.7	9.7	19.5	5.5	136	6113	0.9	0.9
Mahalanobis [21]	X	19.6	29.4	132	147	19.1	19.2	28.9	8.8	124	4009	1.9	2.1
MC dropout [35]	X	4.8	18.1	160	120	3.4	4.3	8.7	14.8	158	1835	0.7	0.9
Ensemble [32]	X	3.1	1.1	162	1643	0.3	0.4	6.6	0.5	156	226622	0.0	0.0
Void classifier [7]	✓	9.2	39.1	149	38	14.6	14.9	9.6	16.6	149	304	6.6	6.6
Embedding density [7]	X	5.9	10.8	155	202	5.3	4.9	12.1	5.7	150	3990	0.7	0.7
Image resynthesis [14]	X	5.1	12.6	157	191	4.4	4.1	6.3	6.0	157	5875	0.3	0.3
SynBoost [39]	✓	27.9	48.6	107	62	40.7	38.0	35.3	16.6	97	723	14.2	13.3
Maximized entropy [11]	✓	21.1	48.6	121	56	33.2	30.0	27.1	12.1	113	1160	7.6	6.9

Table 8: Comparison of benchmark results for the Fishyscapes LostAndFound validation dataset when not using the filtering included in our default segmentation post-processing step. This dataset contains 165 ground-truth components in total. The main performance metrics are highlighted with gray columns.

These datasets show similar scenes and objects as in RoadAnomaly21 test and RoadObstacle21 test, respectively. The splits contain 10 images with 16 ground truth components and 30 images with 45 ground truth objects in total, respectively. Note that although both datasets share the same setup as in the corresponding test splits, they are still **not** representative for the test data since they contains only a very limited number of different road surfaces and diverse obstacle types. Therefore we do not recommend to fine-tune methods on these two validation datasets.

Moreover, we applied our set of anomaly segmentation methods to RoadObstacle21 validation, see [table 11](#). Some of those methods are also made publicly available in our benchmark code to compare to and reproduce the reported results.

F.2 Fishyscapes LostAndFound

The Fishyscapes LostAndFound validation dataset [7] consists of 100 images from the original LostAndFound data [25] with refined labels. With this labeling, anomalous objects are not restricted to only appear on the road but everywhere in the image, therefore Fishyscapes LostAndFound fits our benchmark’s anomaly track.

Comparing the RoadAnomaly21 and Fishyscapes LostAndFound datasets in terms of anomaly class frequency per pixel location, as observed in [figure 10](#), one notices a clear difference in the variation of object locations and sizes. While in Fishyscapes LostAndFound the objects appear mostly in the center of the image and are also rather small, the objects in RoadAnomaly21 may appear everywhere in the image and have sizes ranging from 122 up to 883,319 pixels (thus covering up to more than one third of the image). The low variety in object sizes is also noticeable in the pixel-wise class distribution, as in RoadAnomaly21 13.8% of the pixels belong to the anomaly class and 82.2% to non-anomaly whereas in Fishyscapes LostAndFound only 0.23% belong to anomaly and 81.13% to non-anomaly.

		Component-level metrics with filtering						Component-level metrics without filtering							
Method	OoD data	$k \in \mathcal{K}$		$\hat{k} \in \hat{\mathcal{K}}$		$\tau = 0.50$		$\overline{F_1} \uparrow$	$k \in \mathcal{K}$		$\hat{k} \in \hat{\mathcal{K}}$		$\tau = 0.50$		$\overline{F_1} \uparrow$
		sIoU	PPV	FN	FP	F_1			sIoU	PPV	FN	FP	F_1		
Maximum softmax [23]	X	14.2	62.2	1575	602	11.0	13.4	16.3	17.5	1572	31481	0.8	1.1		
ODIN [22]	X	38.9	48.0	971	1303	39.4	38.1	40.2	29.9	967	5962	17.6	17.2		
Mahalanobis [21]	X	33.8	31.7	1126	2314	25.3	24.6	34.7	22.8	1124	7677	11.7	11.6		
MC dropout [35]	X	17.0	34.7	1453	1641	14.2	14.7	17.7	20.0	1451	9560	4.5	4.7		
Ensemble [32]	X	6.7	7.6	1604	5649	2.8	2.7	7.5	3.8	1600	299431	0.1	0.1		
Void classifier [7]	✓	0.7	35.1	1698	108	1.2	1.1	0.7	25.1	1698	351	1.1	1.0		
Embedding density [7]	X	37.8	35.2	963	1973	33.7	30.8	38.6	18.9	961	6862	16.1	14.8		
Image resynthesis [14]	X	27.2	30.7	1232	2093	22.3	21.5	28.0	19.7	1228	15418	5.5	5.3		
Road inpainting [42]	X	49.2	60.7	749	646	57.9	56.9	50.4	33.0	743	4852	25.7	25.2		
SynBoost [39]	✓	37.2	72.3	930	230	57.3	53.0	37.6	63.3	931	535	51.5	47.7		
Maximized entropy [11]	✓	45.9	63.1	781	598	57.4	55.0	46.7	35.8	778	2813	34.1	32.7		

Table 9: Comparison of benchmark results for the LostAndFound test-NoKnown dataset when not using the filtering included in our default segmentation post-processing step. This dataset contains 1709 ground-truth components in total. The main performance metrics are highlighted with gray columns.

As already discussed in section 4, we observe a less pronounced gap between methods designed for image classification and those specifically designed for anomaly segmentation. A detailed overview of our benchmark results on FishyScapes LostAndFound is given in table 12. In this evaluation, we see that the number of false positive components (relative to the number of ground truth components) over multiple thresholds τ is significantly less than on RoadAnomaly21, shown in Table 2. This holds for all evaluated methods, resulting in relatively strong component-wise performance (compared to SynBoost and maximized entropy). Even Mahalanobis and void classifier report strong results, which is due to similarity of this data set to Cityscapes as all LostAndFound images share the same setup as in Cityscapes. These results further indicate the lack in diversity in FishyScapes LostAndFound. More specifically, the environments of the scenes shown in LostAndFound do not considerably differ to those shown in Cityscapes whereas our RoadAnomaly21 dataset has a wide variety of scenes since all images are gathered from the web, see figure 13.

E.3 LostAndFound test-NoKnown

The LostAndFound dataset [25] shares the same setup as Cityscapes but includes small obstacles on the road. Therefore, this dataset fits our benchmark’s obstacle track. When a model is trained on Cityscapes, the LostAndFound dataset then contains images with objects that have been previously seen and therefore are not anomalies. As most of our methods are designed for anomaly detection, we filtered out all scenes in the LostAndFound test split where the obstacles belong to known classes, *e.g.* children or bicycles, and call this subset LostAndFound test-NoKnown. In this way, the results obtained with our evaluated methods on LostAndFound test-NoKnown and on our RoadObstacle21 dataset are comparable.

Both datasets have obstacles in the same size range. Both RoadObstacle21 and LostAndFound test-NoKnown have 0.12% of the pixels labeled as obstacles, while 39.08% and 15.31% of the pixels belong to not obstacles, respectively. Regarding the object locations in images, the obstacles in RoadObstacle21 are distributed wider over the image than in LostAndFound, as observed in figure 10. This also implies that in RoadObstacle21 the obstacles appear at stronger varying distances. For an illustration as well as of that variation, we refer to figures 15 and 16. Looking at the results in table 13, we observe for LostAndFound test-NoKnown, just as in FishyScapes LostAndFound (table 12), that methods from image classification perform relatively well in comparison to methods designed for anomaly segmentation. This is again due the limited variety of environments, *i.e.* the road surfaces in this dataset. In our RoadObstacle21 dataset, we therefore provide scenes with obstacles on different road surfaces, such as gravel or a road with cracks, see figure 16.

Regarding the dataset size, LostAndFound achieves their high number of images by densely sampling from video sequences. Consequently, some images depict nearly identical scenes (same environment and obstacle combination with the obstacle approximately at the same distance), see *e.g.* Figure 17. In RoadObstacle21 the number of different environment and obstacle combinations is considerably higher due to the wide variety of 31 object types in the dataset. If multiple images depict the same scene, we made sure that the distance to the obstacle (and therefore the size of the obstacle in the image) varies noticeably from image to image, *c.f.* Figure 18.

F.4 LiDAR Guided Small Obstacle Dataset

The third publicly available dataset to which we applied our benchmark suite is the LiDAR guided Small obstacle Segmentation dataset [26], which can be viewed as a reference dataset for our obstacle track. The results corresponding to this dataset are given in [table 14](#). In general, the given set of methods exhibits poor performance on this dataset. More precisely, obstacles are mostly overlooked, *e.g.* SynBoost as best-performing method still misses 1100 of 1203 components in total at the lowest sIoU threshold $\tau = 0.25$. As the LiDAR guided Small obstacle Segmentation dataset rather focuses on the challenge of detecting obstacles via multiple sensors, including LiDAR, the camera images of this dataset are purposely challenging, *e.g.* due to low illumination, blurry images and barely visible obstacles. [Figure 7](#) shows an example of this dataset, which highlights the difficulty of anomaly detection. This dataset can easily be included into our benchmark and it also fits the obstacle track, however, from our experiments we conclude that this dataset is less suitable to camera-only obstacle segmentation as obstacles are not well captured via cameras.

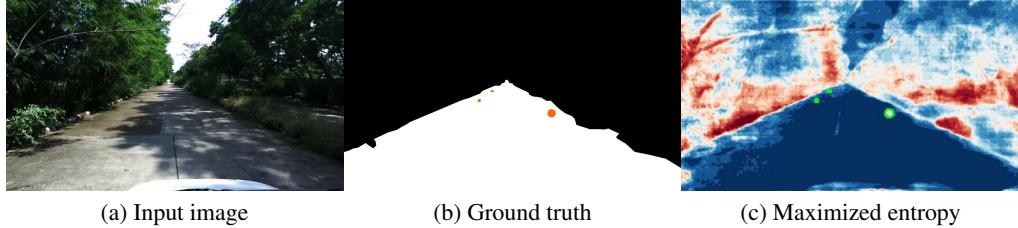


Figure 7: An example image (a) from the Small Obstacle dataset with the corresponding ground truth annotation (b) and an obstacle score heatmap obtained with maximized entropy (c). Here, the obstacles are barely visible in the input image due to their size and the scene’s illumination, that is why camera-only based segmentation techniques tend to fail for this dataset.

F.5 CAOS BDD-Anomaly

The CAOS BDD-Anomaly dataset [8] consists of images sourced from BDD100k [2]. In order to create an anomaly segmentation dataset, the authors split the BDD100k data such that images with motorcycles, bicycles and trains are separated from the rest. These left out objects are then considered as anomalies. We do not perform any experiments on CAOS BDD-Anomaly since the considered anomalous objects are not strictly unknown. They also appear in Cityscapes [1] on which most semantic segmentation models are trained. Moreover, we find several labeling mistakes that hinder proper evaluation of anomaly segmentation performance, see [figure 8](#).

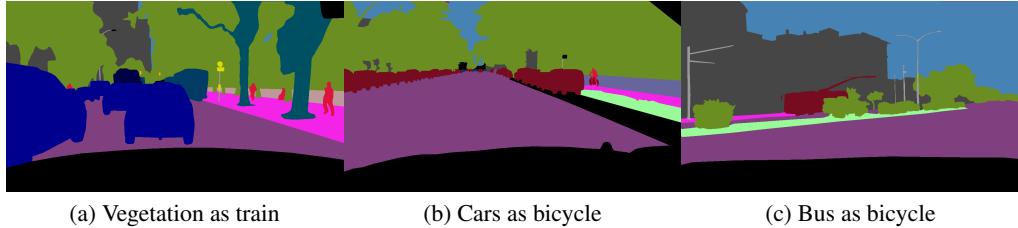


Figure 8: Some examples of labeling mistakes in the CAOS BDD-Anomaly dataset, where in-distribution objects are incorrectly annotated as anomaly, *i.e.* train and bicycle.

G Evaluation per Environment Category

We already emphasized that in our RoadObstacle21 dataset a wide variety of road surfaces are available, representing different scenes which might pose unique challenges. In this section, we provide more insights by evaluating our set of methods on each of these surfaces. In total, we split our datasets into 9 different scenes, shown in [figure 9](#):

1. cracked road, surrounded by snow (road cracked)
2. dark asphalt after rain, with leaves (asphalt dark)
3. gravel road, no snow (road gravel)
4. gray asphalt in village and forest (asphalt gray)
5. motorway with side railing (motorway)
6. sun reflection off wet road (sun reflection)
7. road made of bricks (road bricks)
8. night images (asphalt night)
9. and snowstorm images .



Figure 9: The scenes of our RoadObstacle21 dataset feature a variety of road surfaces.

We evaluate each subset using our benchmark suite and report the results in [table 16](#). This more detailed evaluation shows that the reported set of methods perform differently across the data splits, with no method having consistent performance on each of these subsets. Our dataset offers extra difficulty caused by the diversity of road texture, surrounding environments, weather and lighting variations. Cracks and leaves may trigger false positives, and a gravel or wet road surface may itself be sufficiently different from training images to be mistaken for an anomaly.

H Evaluation for Different Component Sizes

In this section we provide further insights of the segmentation quality of ground truth components in RoadAnomaly21 and RoadObstacle21. To this end, we conduct a more fine-grained analysis by grouping ground truth components into size intervals and perform the evaluation for each size interval separately. In total, RoadAnomaly21 contains 259 ground truth components, ranging in size from 122 to 883,319 pixels. RoadObstacle21 contains 388 obstacles ranging from 18 up to 77,435 pixels. For each dataset we divide these components into eight size intervals such that each interval contains same number of components.

In [figure 11](#), we report the averaged sIoU ([equation \(1\)](#)) w.r.t. the ground truth components within each size interval. As illustrated in this figure, we observe a positive correlation of sIoU with the component size. Especially in RoadObstacle21, methods designed for the task of anomaly segmentation like maximized entropy or SynBoost perform significantly better than the other approaches.

In addition, we consider the amount of entirely neglected components, meaning the objects for which not even one pixel is detected. To do so, we measure the relative ratio of FN to all ground truth components within different object size intervals, see [figure 12](#). As a threshold, therefore, for discriminating between FN and TP, we choose $\tau = 0$, *i.e.* a ground truth component is considered as TP if at least one of its pixels is detected by the respective method. Indeed, we observe a negative correlation of the number of FN with the component size, but even more conspicuous is the amount of totally overlooked components of small size. This analysis shows the challengingness of anomaly segmentation, particularly for small obstacles at component-level, and emphasizes the need for further research in this direction.

I Evaluation per Object Category

As part of our benchmark, we also provide an evaluation with respect to different object categories. An exemplary evaluation with the given set of methods is provided in [table 15](#). In particular, the methods specifically designed for anomaly segmentation perform worse on the vehicle category than on the other ones. This general trends shows that our choice of vehicles, including classes such as jet ski, rickshaw and carriage, is rather challenging. This additional dimension of granularity offers further insight to users of our benchmark such that one can identify the drawbacks of an anomaly segmentation method under inspection.

Tables and Figures

Method			Pixel-level				Component-level									
	requires OoD data		Anomaly scores AuPRC ↑ FPR ₉₅ ↓ F_1^* ↑		$k \in \mathcal{K}$ $\hat{k} \in \hat{\mathcal{K}}$		$\tau = 0.25$		$\tau = 0.50$		$\tau = 0.75$		$\overline{F_1} \uparrow$			
			sIoU ↑	PPV ↑	FN ↓	FP ↓	F_1 ↑	FN ↓	FP ↓	F_1 ↑	FN ↓	FP ↓	F_1 ↑			
Maximum softmax [23]	✗	40.4	60.2	42.6	34.2	18.6	7	37	29.0	10	39	19.7	15	44	3.3	16.7
ODIN [22]	✗	46.3	61.5	49.2	37.1	24.5	6	37	31.7	10	38	20.0	16	47	0.0	16.4
Mahalanobis [21]	✗	22.5	86.4	31.7	17.8	11.7	12	80	8.0	16	84	0.0	16	86	0.0	2.4
MC dropout [35]	✗	29.2	77.9	35.3	26.6	16.7	9	82	13.3	12	84	7.7	16	88	0.0	5.7
Ensemble [32]	✗	16.0	80.0	30.3	20.9	23.1	11	59	12.5	15	63	2.5	15	64	2.5	4.7
Void classifier [7]	✓	39.3	66.1	42.7	25.2	27.4	9	33	25.0	13	34	11.3	16	38	0.0	11.7
Embedding density [7]	✗	51.9	60.0	54.1	48.1	24.4	3	102	19.8	7	105	13.8	15	117	1.5	12.4
Image resynthesis [14]	✗	76.4	20.5	72.0	46.8	25.3	2	66	29.2	9	68	15.4	15	81	2.0	15.0
SynBoost [39]	✓	68.8	30.9	65.6	46.7	21.9	3	65	27.7	9	70	15.1	15	76	2.2	15.7
Maximized entropy [11]	✓	80.7	17.4	74.3	63.6	45.0	1	24	54.5	3	25	48.1	11	29	20.0	41.6

Table 10: Benchmark results for the RoadAnomaly21 validation set. This dataset contains 16 ground truth components.

Method			Pixel-level				Component-level									
	requires OoD data		Anomaly scores AuPRC ↑ FPR ₉₅ ↓ F_1^* ↑		$k \in \mathcal{K}$ $\hat{k} \in \hat{\mathcal{K}}$		$\tau = 0.25$		$\tau = 0.50$		$\tau = 0.75$		$\overline{F_1} \uparrow$			
			sIoU ↑	PPV ↑	FN ↓	FP ↓	F_1 ↑	FN ↓	FP ↓	F_1 ↑	FN ↓	FP ↓	F_1 ↑			
Maximum softmax [23]	✗	43.4	3.8	53.7	40.6	20.9	20	113	27.3	22	113	25.4	32	117	14.9	22.5
ODIN [22]	✗	46.6	4.0	48.4	38.7	28.4	23	84	29.1	23	84	29.1	33	88	16.6	26.2
Mahalanobis [21]	✗	25.9	26.1	27.7	28.3	27.3	24	111	23.7	33	114	14.0	41	120	4.7	14.8
MC dropout [35]	✗	7.9	43.8	13.4	8.9	8.7	39	245	4.1	42	246	2.0	45	246	0.0	2.0
Ensemble [32]	✗	4.7	98.3	9.2	4.1	57.9	42	8	10.7	44	8	3.7	45	8	0.0	4.6
Void classifier [7]	✓	9.8	43.6	15.6	11.2	24.2	39	35	14.0	39	35	14.0	42	36	7.1	11.9
Embedding density [7]	✗	1.5	56.7	3.4	15.9	2.8	31	1261	2.1	39	1262	0.9	45	1269	0.0	1.0
Image resynthesis [14]	✗	70.3	1.3	61.3	28.8	22.4	25	105	23.5	31	108	16.8	43	117	2.4	15.1
Road inpainting [42]	✗	90.4	98.9	89.0	52.9	67.0	17	14	64.4	18	14	62.8	22	16	54.8	61.6
SynBoost [39]	✓	81.4	2.8	73.2	37.0	43.6	18	25	55.7	28	30	37.0	39	37	13.6	38.4
Maximized entropy [11]	✓	94.4	0.4	88.4	56.4	60.8	12	19	68.0	13	19	66.7	24	21	48.3	62.3

Table 11: Benchmark results for the RoadObstacle21 validation set. This dataset contains 45 ground truth objects.

Method			Pixel-level				Component-level									
	requires OoD data		Anomaly scores AuPRC ↑ FPR ₉₅ ↓ F_1^* ↑		$k \in \mathcal{K}$ $\hat{k} \in \hat{\mathcal{K}}$		$\tau = 0.25$		$\tau = 0.50$		$\tau = 0.75$		$\overline{F_1} \uparrow$			
			sIoU ↑	PPV ↑	FN ↓	FP ↓	F_1 ↑	FN ↓	FP ↓	F_1 ↑	FN ↓	FP ↓	F_1 ↑			
Maximum softmax [23]	✗	5.6	40.5	12.1	3.5	9.5	152	199	6.9	164	199	0.5	165	199	0.0	1.8
ODIN [22]	✗	15.5	38.4	22.5	9.9	21.9	139	142	15.6	146	142	11.7	163	143	1.3	9.7
Mahalanobis [21]	✗	32.9	8.7	37.3	19.6	29.4	111	145	29.7	132	147	19.1	155	157	6.0	19.2
MC dropout [35]	✗	14.4	47.8	20.0	4.8	18.1	149	120	10.6	160	120	3.4	164	121	0.7	4.3
Ensemble [32]	✗	0.3	90.4	0.7	3.1	1.1	159	1643	0.7	162	1643	0.3	163	1643	0.2	0.4
Void classifier [7]	✓	11.7	15.3	21.9	9.2	39.1	143	38	19.6	149	38	14.6	158	38	6.7	14.9
Embedding density [7]	✗	8.9	42.2	14.8	5.9	10.8	148	202	8.9	155	202	5.3	163	202	1.1	4.9
Image resynthesis [14]	✗	5.1	29.8	11.1	5.1	12.6	150	190	8.1	157	191	4.4	164	191	0.6	4.1
SynBoost [39]	✓	64.9	30.9	67.6	27.9	48.6	103	62	42.9	107	62	40.7	130	63	26.6	38.0
Maximized entropy [11]	✓	44.3	37.7	50.9	21.1	48.6	117	56	35.7	121	56	33.2	146	57	15.8	30.0

Table 12: Benchmark results for the Fishyscapes LostAndFound validation set. This dataset contains 165 ground truth objects.

Method		Pixel-level	Component-level															
			requires OoD data	Anomaly scores				$k \in \mathcal{K}$		$\hat{k} \in \hat{\mathcal{K}}$		$\tau = 0.25$		$\tau = 0.50$		$\tau = 0.75$		
				AuPRC ↑	FPR ₉₅ ↓	$F_1^* \uparrow$	sIoU ↑	PPV ↑		FN ↓	FP ↓	$F_1 \uparrow$	FN ↓	FP ↓	$F_1 \uparrow$	FN ↓	FP ↓	$F_1 \uparrow$
Maximum softmax [23]	x	30.1	33.2	32.5	14.2	62.2	1256	580	33.0	1575	602	11.0	1701	644	0.7	13.4		
ODIN [22]	x	51.0	30.7	54.3	38.9	48.0	713	1271	50.1	971	1303	39.4	1349	1372	20.9	38.1		
Mahalanobis [21]	x	55.0	12.9	54.8	33.8	31.7	777	2146	38.9	1126	2314	25.3	1527	2585	8.1	24.6		
MC dropout [35]	x	36.2	36.0	42.2	17.0	34.7	1214	1626	25.8	1453	1641	14.2	1635	1674	4.3	14.7		
Ensemble [32]	x	2.9	82.0	8.2	6.7	7.6	1523	5633	4.9	1604	5649	2.8	1695	5705	0.4	2.7		
Void classifier [7]	✓	4.4	47.0	13.7	0.7	35.1	1689	108	2.2	1698	108	1.2	1708	109	0.1	1.1		
Embedding density [7]	x	61.7	10.4	61.7	37.8	35.2	646	1873	45.8	963	1973	33.7	1526	2299	8.7	30.8		
Image resynthesis [14]	x	57.1	8.8	55.1	27.2	30.7	947	1990	34.2	1232	2093	22.3	1560	2304	7.2	21.5		
Road inpainting [42]	x	83.0	35.7	79.1	49.2	60.7	631	635	63.0	749	646	57.9	958	727	47.1	56.9		
SynBoost [39]	✓	81.8	4.6	75.2	37.2	72.3	767	203	66.0	930	230	57.3	1378	436	26.7	53.0		
Maximized entropy [11]	✓	77.9	9.7	76.8	45.9	63.1	639	589	63.5	781	598	57.4	1113	681	39.9	55.0		

Table 13: Benchmark results for the LostAndFound test-NoKnown dataset. This dataset contains 1709 ground truth objects.

Method		Pixel-level	Component-level															
			requires OoD data	Anomaly scores				$k \in \mathcal{K}$		$\hat{k} \in \hat{\mathcal{K}}$		$\tau = 0.25$		$\tau = 0.50$		$\tau = 0.75$		
				AuPRC ↑	FPR ₉₅ ↓	$F_1^* \uparrow$	sIoU ↑	PPV ↑		FN ↓	FP ↓	$F_1 \uparrow$	FN ↓	FP ↓	$F_1 \uparrow$	FN ↓	FP ↓	$F_1 \uparrow$
Maximum softmax [23]	x	0.7	57.1	2.2	0.5	1.5	1196	1652	0.5	1202	1653	0.1	1203	1653	0.0	0.2		
ODIN [22]	x	1.7	51.7	5.7	2.7	3.9	1151	1829	3.4	1176	1834	1.8	1197	1841	0.4	1.9		
Mahalanobis [21]	x	1.4	45.5	2.4	7.1	4.0	1039	4863	5.3	1137	4882	2.1	1198	4907	0.2	2.4		
MC dropout [35]	x	0.5	82.2	2.1	0.5	2.8	1191	1406	0.9	1200	1407	0.2	1203	1408	0.0	0.3		
Void classifier [7]	✓	0.8	59.6	2.1	1.5	4.9	1169	813	3.3	1193	816	1.0	1200	819	0.3	1.5		
Embedding density [7]	x	0.5	66.0	1.1	9.8	1.8	1010	12421	2.8	1122	12502	1.2	1200	12587	0.0	1.3		
SynBoost [39]	✓	12.5	62.8	22.8	11.5	14.4	1009	1204	14.9	1040	1217	12.6	1116	1234	6.9	12.0		
Maximized entropy [11]	✓	4.9	63.1	11.6	2.0	9.7	1159	586	4.8	1184	586	2.1	1202	586	0.1	2.4		

Table 14: Benchmark results for the LiDAR guided Small obstacle Segmentation dataset. This dataset contains 1203 ground truth components in total.

Method		all anomalies				animals				vehicles				other anomalies			
		OoD data	N = 100			OoD data	N = 59			OoD data	N = 23			OoD data	N = 11		
			AuPRC ↑	FPR ₉₅ ↓	$F_1 \uparrow$		AuPRC ↑	FPR ₉₅ ↓	$F_1 \uparrow$		AuPRC ↑	FPR ₉₅ ↓	$F_1 \uparrow$		AuPRC ↑	FPR ₉₅ ↓	$F_1 \uparrow$
Maximum softmax [23]	x	28.0	72.0	5.4	25.2	75.6	4.9	30.2	71.9	5.0	26.1	57.7	9.3				
ODIN [22]	x	31.1	71.7	5.2	32.1	72.9	4.9	30.6	74.0	4.9	35.5	61.7	9.7				
Mahalanobis [21]	x	20.0	87.0	2.7	21.3	87.4	2.5	16.7	87.5	1.8	34.9	66.1	12.5				
MC dropout [35]	x	28.9	69.5	4.3	24.8	74.0	3.2	35.2	72.2	4.5	20.1	62.3	15.2				
Ensemble [32]	x	17.7	91.1	3.4	16.7	91.3	2.9	18.8	89.7	1.1	10.4	85.6	4.8				
Void classifier [7]	✓	36.6	63.5	6.5	32.2	66.9	4.0	42.3	39.2	8.5	23.1	70.3	21.7				
Embedding density [7]	x	37.5	70.8	7.9	43.9	63.2	8.4	30.3	88.4	3.4	24.2	58.4	21.0				
Image resynthesis [14]	x	52.3	25.9	12.5	51.4	26.5	16.4	57.8	25.6	6.1	40.4	55.1	12.9				
SynBoost [39]	✓	56.4	61.9	10.0	54.7	66.2	10.3	57.8	61.7	7.4	43.1	62.6	21.4				
Maximized entropy [11]	✓	85.5	15.0	28.7	92.2	7.2	41.9	79.0	17.8	16.2	51.6	18.3	25.4				

Table 15: Effect of different anomalies in the RoadAnomaly21 dataset. In total, RoadAnomaly21 contains 59 images with only animals, 23 images with only vehicles and 11 with other anomalies (e.g. cones, tents, . . .). The number of images according to a subset is denoted with N in the table. Images containing objects from both the animal and the vehicle category (7 images in total) are excluded in this evaluation.

		road cracked	asphalt dark	road gravel	asphalt gray	motorway	sun reflection	road bricks	
Method	OoD	$N = 40$	$N = 47$	$N = 33$	$N = 66$	$N = 30$	$N = 72$	$N = 39$	
	data	AuPRC	$\overline{F_1}$	AuPRC	$\overline{F_1}$	AuPRC	$\overline{F_1}$	AuPRC	$\overline{F_1}$
Maximum softmax [23]	\times	11.7	3.2	69.3	25.7	39.5	21.0	43.4	14.9
ODIN [22]	\times	14.9	4.8	74.8	30.8	65.3	37.0	73.8	22.5
Mahalanobis [21]	\times	25.9	1.6	46.7	18.3	65.8	21.9	84.7	53.8
MC dropout [35]	\times	6.5	1.0	21.7	4.9	27.9	5.5	11.4	1.7
Ensemble [32]	\times	34.3	0.0	5.6	0.8	33.4	0.0	17.3	12.3
Void classifier [7]	\checkmark	15.9	6.4	35.0	15.4	6.3	3.1	38.2	11.0
Embedding density [7]	\times	2.5	0.8	3.3	0.8	2.7	2.2	1.8	2.4
Image resynthesis [14]	\times	48.2	9.6	42.0	12.3	77.0	42.4	66.6	22.2
Road inpainting [42]	\times	21.0	18.4	77.0	47.2	88.4	74.7	93.5	79.8
SynBoost [39]	\checkmark	46.1	14.7	89.3	66.5	84.7	54.5	81.2	54.0
Maximized entropy [11]	\checkmark	77.1	42.5	96.9	71.9	98.6	88.7	94.8	70.2

		asphalt night		snowstorm	
Method	OoD	$N = 30$	$N = 55$		
	data	AuPRC	$\overline{F_1}$	AuPRC	$\overline{F_1}$
Maximum softmax [23]	\times	6.0	2.5	1.6	0.8
ODIN [22]	\times	8.0	1.8	6.7	4.6
Mahalanobis [21]	\times	14.2	5.5	21.2	13.2
MC dropout [35]	\times	4.2	1.1	0.5	0.6
Ensemble [32]	\times	11.5	16.9	0.6	0.0
Void classifier [7]	\checkmark	5.9	5.5	3.0	5.1
Embedding density [7]	\times	16.7	3.6	0.9	2.6
Image resynthesis [14]	\times	16.5	6.3	19.2	4.0
Road inpainting [42]	\times	51.2	28.0	55.3	35.0
SynBoost [39]	\checkmark	14.5	10.2	46.4	20.7
Maximized entropy [11]	\checkmark	41.0	12.1	30.5	17.5

Table 16: Effect of different scenes in the RoadObstacle21 dataset. Here, N denotes the number of images in a subset. As main evaluation metrics we consider the pixel-wise AuPRC and the component-wise \overline{F}_1 .

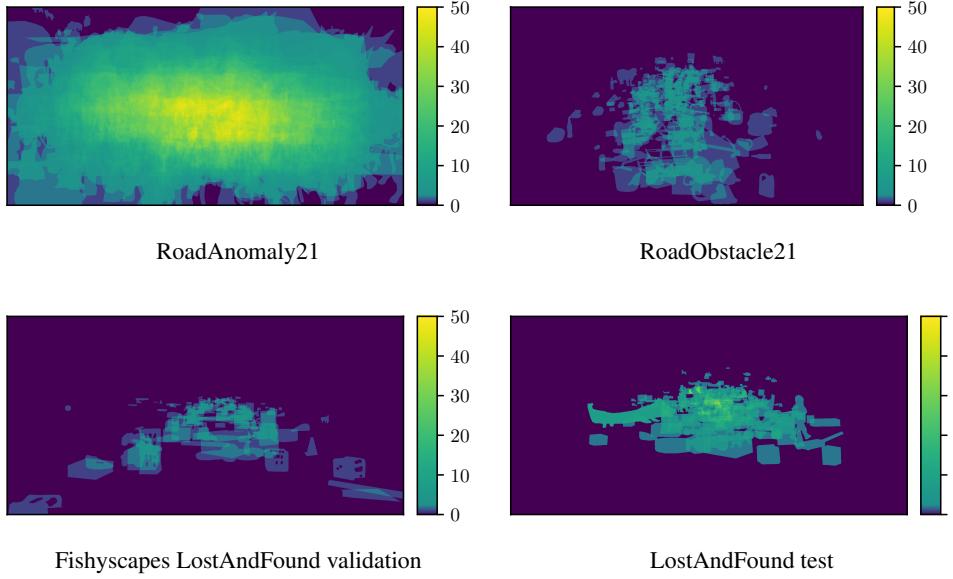
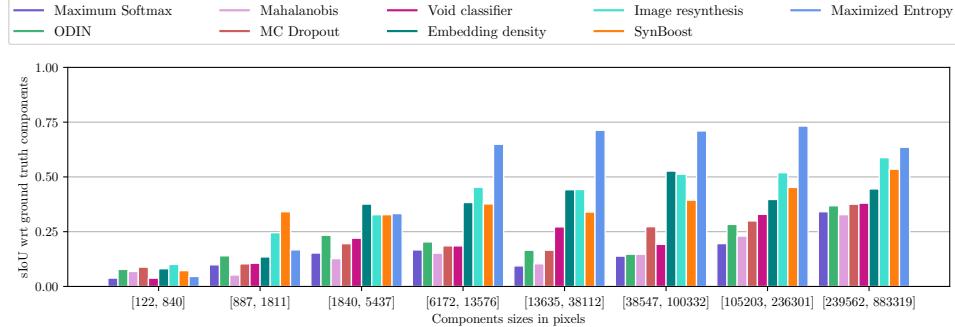
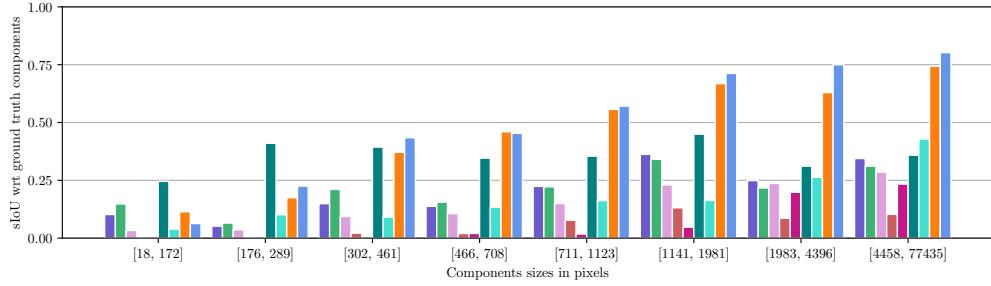


Figure 10: Comparison of the (spatial) pixel distributions between RoadAnomaly21 and Fishscapes LostAndFound (100 images each) as well as RoadObstacle21 and a subset of randomly sampled images from the LostAndFound test dataset (327 images each). The color indicates the frequency of observing an anomaly in each pixel location, averaged over the images in the dataset.

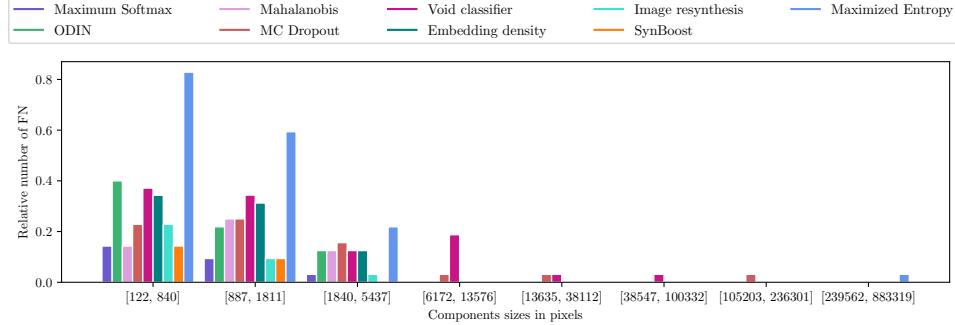


RoadAnomaly21, each size interval contains 32 components except the first one (very left) that contains 35 components

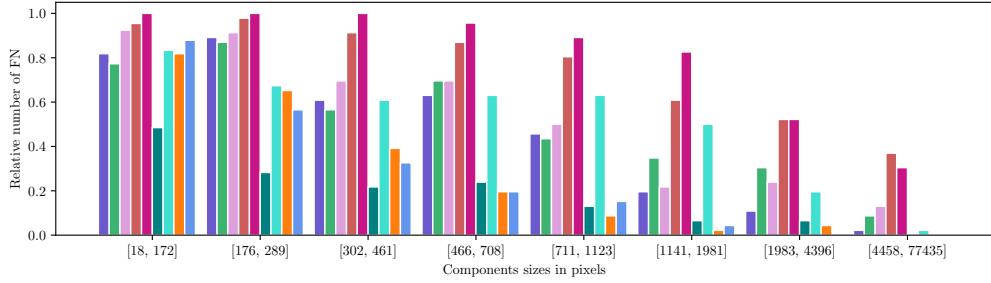


RoadObstacle21, each size interval contains 46 components except the first one (very left) that contains 66 components

Figure 11: Comparison of the averaged sIoU w.r.t. ground truth components within a certain range of the components size, produced by the methods discussed in section 3.3 and appendix D.2.



RoadAnomaly21, each size interval contains 32 components except the first one (very left) that contains 35 components



RoadObstacle21, each size interval contains 46 components except the first one (very left) that contains 66 components

Figure 12: Comparison of the relative number of FN to TP at threshold $\tau = 0$, i.e. the fraction of overlooked components to the total number of ground truth components within a certain range of the components size. The evaluated methods are discussed in section 3.3 and appendix D.2.

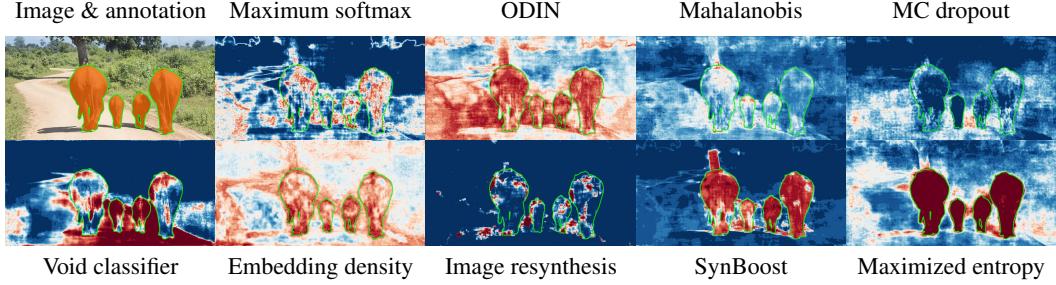


Figure 13: Qualitative comparison of the methods introduced in section 3.3 and appendix D.2 on a sample from RoadAnomaly21. In this example, the anomalous objects have a large size and the environment differs from scenes shown in Cityscapes. Green contours indicate the annotation of the anomaly.

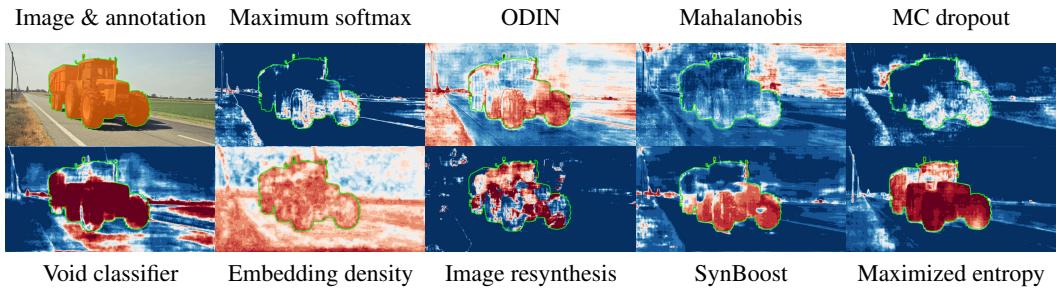


Figure 14: Qualitative comparison of the methods introduced in section 3.3 and appendix D.2 on a sample from RoadAnomaly21. The scene shows a tractor which does not appear in Cityscapes. Green contours indicate the annotation of the anomaly.

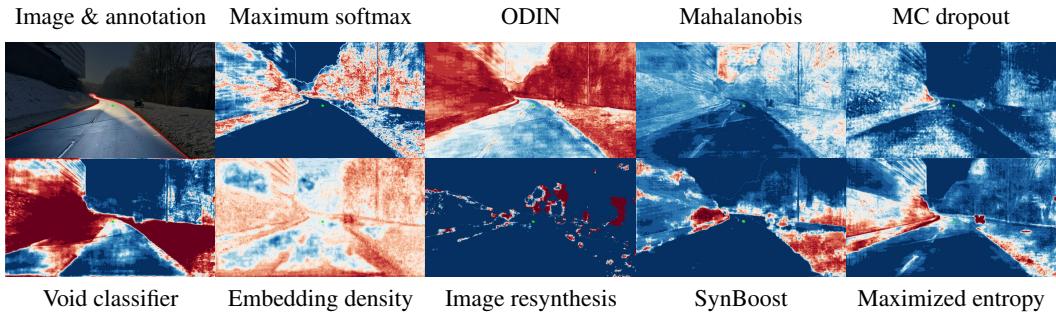


Figure 15: Qualitative comparison of the methods introduced in section 3.3 and appendix D.2 for an example from RoadObstacle21, where the obstacle is small and far away. Green contours indicate the annotation of the obstacle, red contours the road.

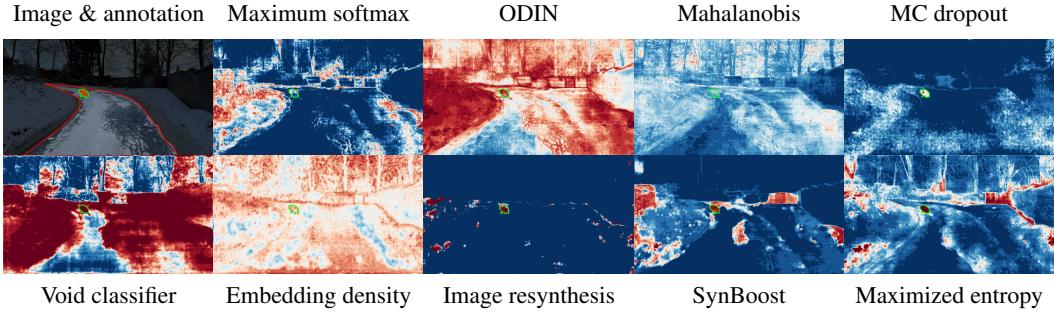


Figure 16: Qualitative comparison of the methods introduced in section 3.3 and appendix D.2 for an example from RoadObstacle21, showing a road surface with cracks. Green contours indicate the annotation of the obstacle, red contours the road.

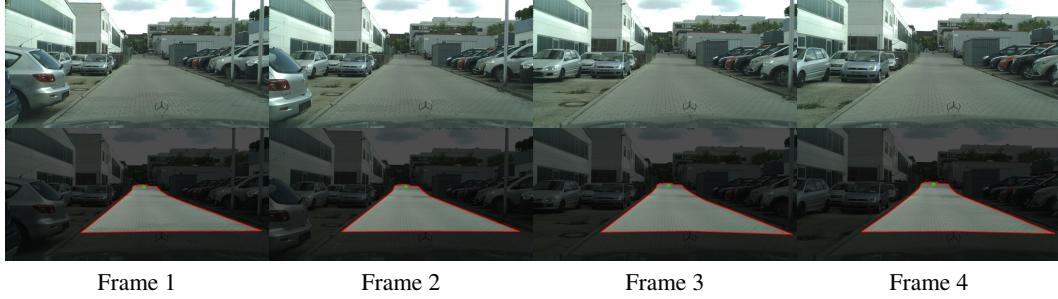


Figure 17: Four example images of densely sampled frames from a video sequence (of 18 frames in total) with ground truth annotation in the LostAndFound test set. Due to this sampling, LostAndFound achieve their high number of images but, as shown in this figure, several images are nearly identical.

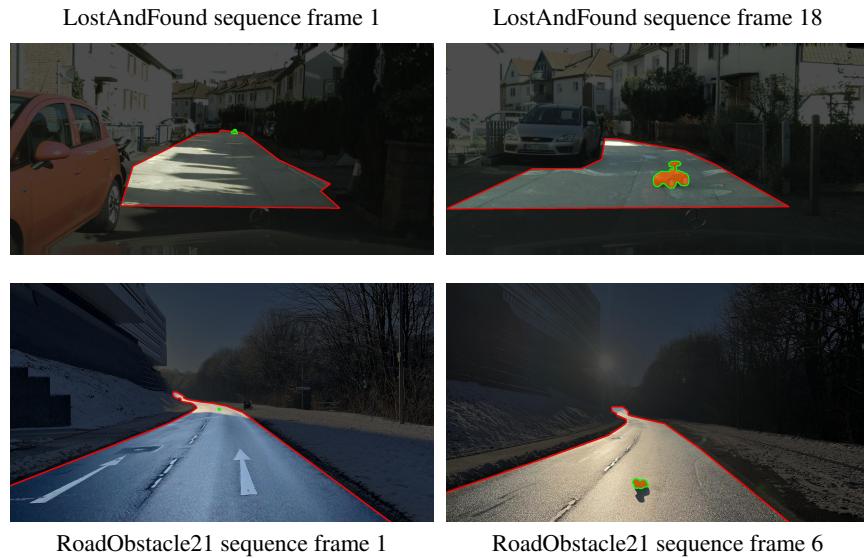


Figure 18: Comparison of one sequence from the LostAndFound test set (top row) and one sequence from the RoadObstacle21 test set (bottom row). In this figure, the first and last frame of a video sequence which are included in the respective test set, are shown. We observe that in this LostAndFound example 18 images of one sequence are included in the test while in RoadObstacle21 at most 6 frames are included (which differ significantly in lighting in this example).