Bergische Universität Wuppertal

Fakultät für Mathematik und Naturwissenschaften

Institute of Mathematical Modelling, Analysis and Computational Mathematics (IMACM)

Preprint BUW-IMACM 21/13

Robin Chan, Krzysztof Lis, Svenja Uhlemeyer, Hermann Blum, Sina Honari, Roland Siegwart, Pascal Fua, Mathieu Salzmann, Matthias Rottmann

SegmentMeIfYouCan: A Benchmark for Anomaly Segmentation

April 30, 2021

http://www.imacm.uni-wuppertal.de

SegmentMeIfYouCan: A Benchmark for Anomaly Segmentation

Robin Chan[*1] Krzysztof Lis[*2] Svenja Uhlemeyer[*1] Hermann Blum[*3]
Sina Honari[2] Roland Siegwart[3] Pascal Fua[2] Mathieu Salzmann[2] Matthias Rottmann[1]

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 specifically designed for anomaly / obstacle segmentation, on our datasets as well as on public ones, using our benchmark suite. The anomaly and obstacle segmentation results show that our datasets contribute to the diversity and challengingness of both dataset landscapes.

1. Introduction

The advent of high-quality publicly-available datasets, such as Cityscapes [13], BDD100k [45], A2D2 [17] and COCO [30] have 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* [47]. Instead, they tend to be overconfident in their predictions, even in the presence of previously-unseen objects [19], which precludes the use of uncertainty to identify the corresponding anomalous regions.

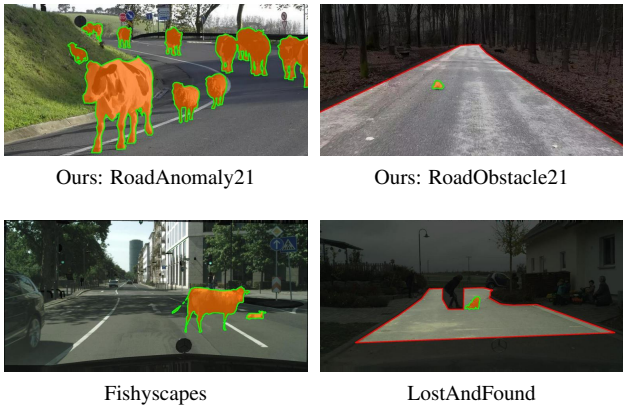


Figure 1: Comparison of images from our and existing public datasets. Anomalies / obstacles are highlighted in orange, darkened regions are excluded from the evaluation. In RoadAnomaly21, anomalies may appear everywhere in the image. In contrast to Fishyscapes, where anomalous objects are synthetic, 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.

Nonetheless, 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 [6] or CAOS [20]. While these benchmarks have enabled interesting experiments, the limited real-world diversity in Fishyscapes, the lack of a public leader board and of a benchmark suite in CAOS, and the reliance on synthetic images in both works hinder proper evaluation and comparisons with the state-of-the-art.

[*]equal contribution
[1]Stochastics Group, IZMD, BUW, Wuppertal, Germany
[2]Computer Vision Laboratory, EPFL, Lausanne, Switzerland
[3]Autonomous Systems Lab, ETHZ, Zürich, Switzerland

In this paper, motivated by the limitations of existing anomaly and obstacle segmentation datasets and by the emerging body of works in this direction [1, 6, 8, 9, 23, 24, 32, 34, 39], we introduce the *SegmentMeIfYouCan*[1] 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 as well as comparisons, to facilitate the development of road anomaly and obstacle 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 it be from a known class or from an unknown one.

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 download[1], 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 numerical experiments that compare benchmark results for a number of state-of-the-art road anomaly / obstacle segmentation methods on our datasets and on other related ones.

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 [12, 35, 43], NotMNIST [43], or Omniglot [26], and separating CIFAR-10 from SVHN [28, 35, 43] or LSUN [28, 29, 35]. Such experiments can be found in many works, including [12, 21, 28, 29, 35, 43].

For semantic segmentation, a similar task was therefore proposed by the WildDash benchmark [46] 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. Similarly, the Robust Vision Challenge[2] evaluated the generalization of segmentation models across different datasets. Here, 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.

One prominent dataset tackling the task of anomaly segmentation is the LostAndFound dataset [40], which shares the same setup as Cityscapes [13] 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. While this was filtered and refined in Fishyscapes [6], the low diversity of anomalies persists.

The CAOS benchmark [20] suffers from a similar low-diversity issue, arising from its use of only 2 object classes from the BDD100k dataset [45] as anomalies. Both Fishyscapes 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. Additional efforts in this regard have been made by sourcing and annotating images of animals in street scenes [32] and a multi-modal dataset of small objects placed on the road [42]. Parts of these datasets have been included in our benchmark. Moreover, most of the above datasets are fully published with annotations, allowing methods to overfit on the available anomalies. Apart from Fishyscapes, which as mentioned above suffers from low diversity, we did not find any public leader boards that allow for a trustworthy comparison of new methods.

2.2. Anomaly and Obstacle Segmentation

In the following, we give an overview of anomaly segmentation methods. For each type, we evaluate at least one method in our experiments. Anomaly detection was initially tackled in the context of image classification, by developing post-processing techniques aiming to adjusting the confidence values produced by a classification model [19, 21, 28, 29, 35]. Although originally designed for image-level anomaly detection, most of these methods can easily

[1] http://www.segmentmeifyoucan.com/

[2] http://www.robustvision.net/rvc2018.php

be adapted to anomaly segmentation [1, 6] by treating each single pixel in an image as a potential anomaly.

Relevant are also works that estimate uncertainty of predictions as anomalous image regions are expected to correlate with high uncertainty. One approach to this is Bayesian (deep) learning [33, 38] where model parameters are treated as distributions. Due to the computational complexity, approximations to Bayesian inference have been developed [2, 16, 18, 27] and extended to semantic segmentation [3, 25, 36]. Instead of reasoning about uncertainty, other, non-Bayesian approaches tune a previously-trained model to the task of anomaly detection, by either modifying its architecture or exploiting additional data. For example, in [15], anomaly scores are learned by adding a separate branch to the neural network. Instead of modifying the network's architecture, other approaches [22, 35] 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 [4, 9, 24].

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 that of unknown regions. 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 [5,32,44] but also specifically for obstacle detection [14, 31, 37].

3. Benchmark Description

In this section we present our datasets and benchmark tracks together with the evaluation metrics.

3.1. Motivation

The aim of our benchmark is on the hand providing diverse data with high-quality and consistent annotations, to facilitate advances in general anomaly segmentation research. On the other hand, by focusing on road scenes, the benchmark should accelerate and measure progress towards the imminent and practical need for safe segmentation methods in automated driving.

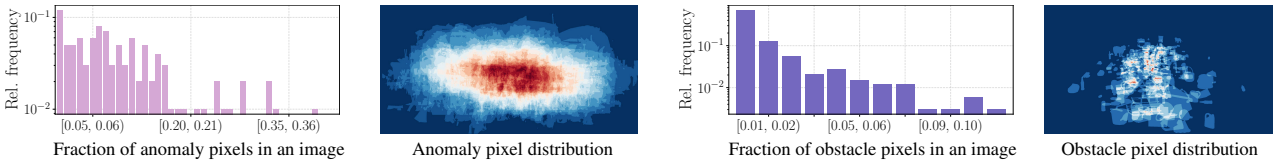
To achieve these goals, our benchmark covers two tasks. First, it tackles the general problem of anomaly segmentation, aiming to identify the image regions containing objects that have never been seen during training, and thus for which segmentation is not trustworthy. This is necessary for any reliable decision making process, and of great importance to many computer vision applications. This definition of anomalies, however, can sometimes be ill-defined because (i) existing semantic segmentation datasets, such as Cityscapes [13], 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 or an unknown one.

3.2. 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.

RoadAnomaly21. The road anomaly track benchmarks general anomaly segmentation in full scenes, and is thus independent of automated driving. It consists of an evaluation dataset of 100 images with pixel-level annotations. The data is an extension of that introduced in [32], now including a broader collection of images and finer-grain labeling. In particular, we removed labeling mistakes and added the void class. Each image contains at least one anomalous object, *e.g.* animals or unknown vehicles. The anomalies can appear anywhere in the image and widely differ in size, covering from 0.5% to 40% of the image. The distribution of object sizes is shown in figure 2. 13.8% of the dataset's pixels belong to anomalies and 82.2% to non-anomalies. The images were collected from web resources and therefore depict a wide variety of environments.

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 hazardous obstacles that are critical to detect. This focuses the task of distinguishing between road surface and obstacles. Our dataset consists of 222 new images taken by ourselves and 105 from [31] summing up to a total of 327 evaluation images with pixel-level annotations, where 0.12% of the all pixels are annotated as obstacles and 39.1% as drivable area. 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 anomaly objects as well, *e.g.* stuffed toys, sleighs or tree stumps. They appear at different distances (one distance per image) and are completely surrounded by road pixels. This allows us to focus our evaluation on the obstacles, as other objects lie outside the region of interest. Moreover, this dataset incorporates different road surfaces, lighting and weather conditions, thus encompassing a broad diversity of scenes. An extra track of additional 85



(a) RoadAnomaly21 (b) RoadObstacle21

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

images with scenes at night and in extreme weather, like snowstorm, is also available. However, the latter subset is excluded from our numerical experiments due to the heavy shift in domain.

Labeling Policy. In both datasets the pixel-level annotations include three classes: 1) anomaly / obstacle, 2) not anomaly / not obstacle, and 3) void. The 19 Cityscapes evaluation classes [13], on which many semantic segmentation DNNs are trained, serve as basis to judge whether an object is anomalous or not. We assigned image regions to the void class if they cannot be clearly assigned to any of the Cityscapes classes and also do not belong to the objects / regions of interest, *i.e*. they are neither anomaly nor obstacle (depending on the track). For instance, in the anomaly track, this includes water, trash cans or street lights. Ambiguous cases (*e.g*. constituting a strong domain shift) were labeled as void as well. In the obstacle track, all regions aside the drivable area are assigned to the void class. The void regions are ignored in the evaluation.

3.3. Metrics

For the sake of brevity, in what follows we only refer to anomalies instead of anomalies and obstacles.

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 ($c_1 =$ “anomaly”) we compute

$$\text{precision} = \frac{|\mathcal{Y}_{c_1} \cap \hat{\mathcal{Y}}_{c_1}(\delta)|}{|\hat{\mathcal{Y}}_{c_1}(\delta)|}, \quad \text{recall} = \frac{|\mathcal{Y}_{c_1} \cap \hat{\mathcal{Y}}_{c_1}(\delta)|}{|\mathcal{Y}_{c_1}|} \quad (1)$$

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 [7]. 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 3.2.

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

$$\text{FPR}_{95} = \frac{|\hat{\mathcal{Y}}_{c_1}(\delta') \cap \mathcal{Y}_{c_2}|}{|\mathcal{Y}_{c_2}|} \quad \text{s.t. TPR}(\delta') = 0.95\ , \quad (2)$$

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 3.2. In particular for the RoadObstacle21 dataset the evaluation is therefore restricted to the road area.

Component level. From a practitioner's perspective, it is often sufficient to correctly identify only a fraction of an anomalous object instead of every single pixel. It is however very important in practice to detect all anomalies in the scene, regardless of their number of pixels. Thus, to evaluate how well a model performs at localizing anomalies, we consider performance metrics 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 [41]. In particular, while in [41] the sIoU is

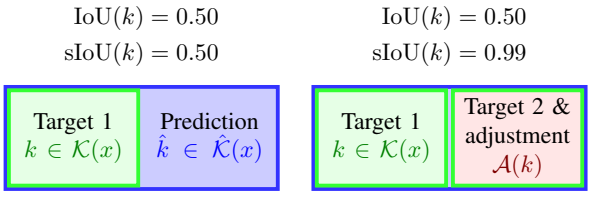


Figure 3: Illustration of the adjusted component-wise intersection over union. Here, IoU denotes the ordinary component-wise intersection over union, sIoU the adjusted one. In both examples the prediction $\hat{k}$ (blue rectangle) is the same but covers different targets (green rectangles). In the sIoU the set $\mathcal{A}(k)$ is subtracted from the union (in this case equal to $\hat{k}$) when evaluating target k with respect to prediction $\hat{k}$ (right figure). That is, the sIoU does not penalize the case when one prediction covers multiple targets. Since two targets are separated by at least one pixel, IoU = sIoU = 1 iff the prediction covers one target perfectly.

computed for predicted components only (to determine FP), we also consider the sIoU for ground-truth components to compute TP and FN. We discuss this in 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 $\mathrm{sIoU} : \mathcal{K} \dot{\cup} \hat{\mathcal{K}} \to [0, 1]$. For $k \in \mathcal{K}$, it is defined as

$$\mathrm{sIoU}(k) := \frac{|k \cap \hat{K}(k)|}{|k \cup \hat{K}(k)| - |\mathcal{A}(k)|} \quad \text{with} \quad \hat{K}(k) = \bigcup_{\substack{\hat{k} \in \hat{\mathcal{K}} \\ \hat{k} \cap k \neq \emptyset}} \hat{k} \tag{3}$$

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 $\mathrm{sIoU}(k) > \tau$, and FN otherwise.

For the other error type, *i.e.* FP, we consider $\hat{k} \in \hat{\mathcal{K}}$ in equation (3), and compute

$$\mathrm{sIoU}(\hat{k}) := \frac{|\hat{k} \cap K(\hat{k})|}{|\hat{k} \cup K(\hat{k})| - |\mathcal{A}(\hat{k})|}, \tag{4}$$

with analogous definitions of $K(\hat{k})$ and $\mathcal{A}(\hat{k})$. We then call a predicted component $\hat{k} \in \hat{\mathcal{K}}$ FP if $\mathrm{sIoU}(\hat{k}) \leq \tau$.

As overall metric for component-level evaluation, we include the component-wise F_1-score that is defined as

$$F_1(\tau) := \frac{2 \cdot \mathrm{TP}(\tau)}{2 \cdot \mathrm{TP}(\tau) + \mathrm{FN}(\tau) + \mathrm{FP}(\tau)} \in [0, 1], \tag{5}$$

and summarizes the TP, FN and FP quantities (which depend on τ). The component-level metrics allow evaluating localization of objects irrespective of their size and hence big objects will not dominate these metrics. In addition, while object detection metrics punish if a prediction covers multiple ground truth objects or vice-versa, our component-level metric does not do so, see figure 3.

4. Evaluated Methods

In this section, we briefly describe the methods which are evaluated on our benchmark and constitute our initial leader board. All methods subject to evaluation are stated in boldface. 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. For additional technical details, we refer the reader to the supplementary material.

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 [21]. 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** [29]. 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 [36] and use the mutual information as pixel-wise anomaly scores, which captures the epistemic uncertainty of a DNN.

In [6] 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.

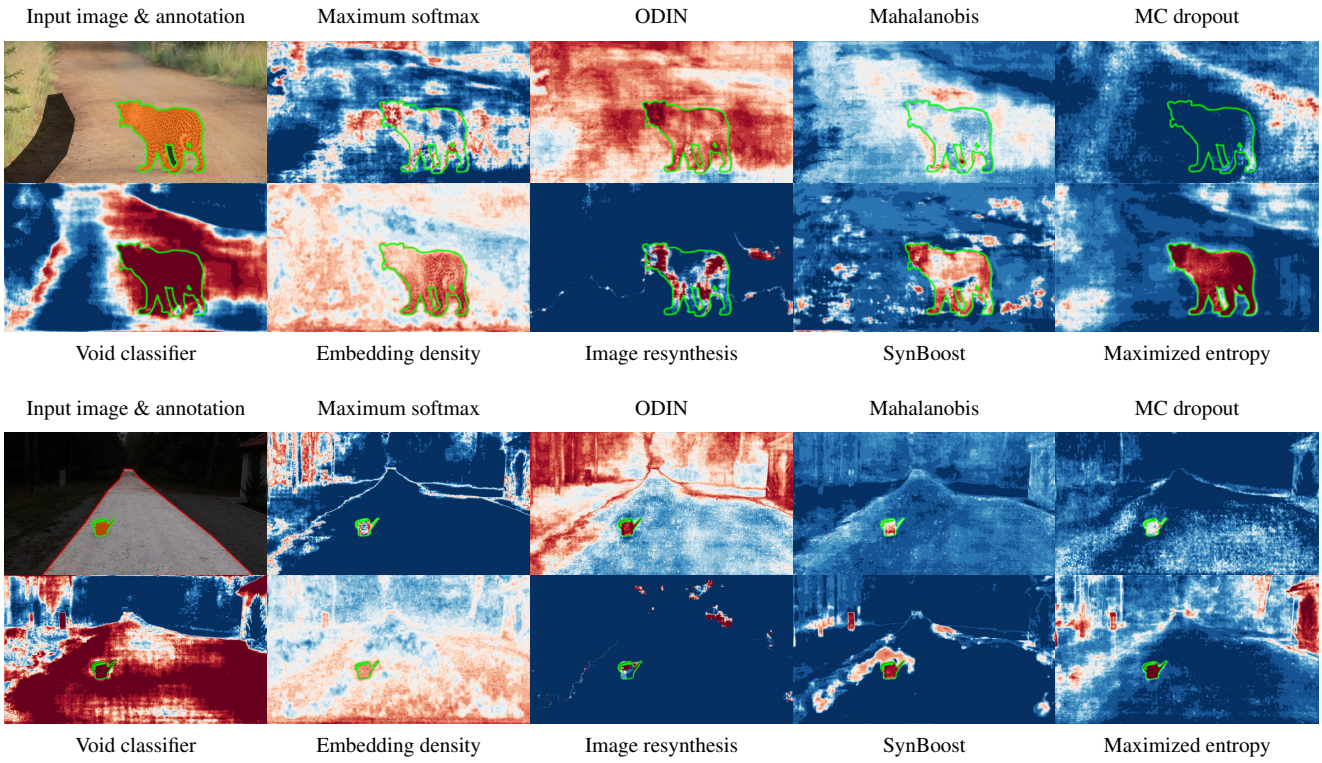


Figure 4: Qualitative comparison of the anomaly scores produced by the methods introduced in section 4 for one example image of RoadAnomaly21 (top two rows) and one example image of RoadObstacle21 (bottom two rows). Here, red indicates higher anomaly / obstacle scores. The ground truth anomaly / obstacle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

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 [6], 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 [30], and enforce **maximized softmax entropy** [9] on samples of the OoD proxy. All theses 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 [32]. This approach can be extended by including uncertainty estimates in the discrepancy module, aiming to boost the anomaly segmentation performance, known as **SynBoost** [5]. One method specifically designed for obstacle segmentation is called **road inpainting** [31]. 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 [32], for pixel-wise obstacle scores.

5. Numerical Experiments

In our benchmark suite we integrate a default method to generate the 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

$$\delta^* = \arg\max_{\delta \in \mathbb{R}} \frac{2 \cdot \text{precision}(\delta) \cdot \text{recall}(\delta)}{\text{precision}(\delta) + \text{recall}(\delta)} \tag{6}$$

subject to $\text{precision}(\delta) + \text{recall}(\delta) > 0, \; \forall \, \delta$.

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 instances. All methods presented in section 4 produce anomaly scores for which we apply the default segmentation method. Note that, we also allow competitors in the benchmark to submit anomaly segmentation masks generated via more sophisticated operations.

Method	requires OoD data	Pixel-level: Anomaly scores AuPRC ↑	FPR$_{95}$ ↓	F_1^* ↑	Component-level: $k \in \mathcal{K}$ $\overline{\text{sIoU}}$ ↑	$\hat{k} \in \hat{\mathcal{K}}$ $\overline{\text{sIoU}}$ ↑	sIoU $> \tau = 0.25$ FN ↓	FP ↓	F_1 ↑	sIoU $> \tau = 0.50$ FN ↓	FP ↓	F_1 ↑	sIoU $> \tau = 0.75$ FN ↓	FP ↓	F_1 ↑	$\overline{F_1}$ ↑
Maximum softmax [21]	✗	27.9	72.2	34.0	15.2	4.7	204	759	10.3	232	789	5.0	253	811	1.1	5.2
ODIN [29]	✗	33.4	72.0	39.3	20.1	4.2	182	1142	10.4	224	1182	4.7	250	1209	1.2	5.2
Mahalanobis [28]	✗	20.1	86.9	31.9	15.0	2.5	203	1581	5.9	238	1619	2.2	254	1636	0.5	2.7
MC dropout [36]	✗	29.1	69.9	39.0	20.9	3.3	170	1608	9.1	219	1648	4.1	251	1679	0.8	4.4
Void classifier [6]	✓	36.9	63.4	44.3	21.3	5.5	177	989	12.3	216	1035	6.4	250	1067	1.3	6.5
Embedding density [6]	✗	37.6	70.8	48.7	34.5	5.0	102	1781	14.3	172	1837	8.0	245	1911	1.3	7.9
Image resynthesis [32]	✗	52.3	25.9	60.5	39.5	6.9	94	1242	19.8	152	1285	13.0	227	1357	3.9	12.4
SynBoost [5]	✓	56.6	62.3	58.2	35.2	6.5	109	1264	17.9	175	1320	10.1	241	1388	2.2	10.2
Maximized entropy [9]	✓	85.6	14.9	77.5	49.2	17.8	85	552	35.3	113	572	29.9	159	611	20.6	29.0

Table 1: Benchmark results for our RoadAnomaly21 dataset. This dataset contains 259 ground truth components in total.

Method	requires OoD data	Pixel-level: Anomaly scores AuPRC ↑	FPR$_{95}$ ↓	F_1^* ↑	Component-level: $k \in \mathcal{K}$ $\overline{\text{sIoU}}$ ↑	$\hat{k} \in \hat{\mathcal{K}}$ $\overline{\text{sIoU}}$ ↑	sIoU $> \tau = 0.25$ FN ↓	FP ↓	F_1 ↑	sIoU $> \tau = 0.50$ FN ↓	FP ↓	F_1 ↑	sIoU $> \tau = 0.75$ FN ↓	FP ↓	F_1 ↑	$\overline{F_1}$ ↑
Maximum softmax [21]	✗	15.7	16.6	22.5	19.7	4.4	255	1673	12.1	326	1738	5.7	372	1783	1.5	6.3
ODIN [29]	✗	22.1	15.3	30.1	21.4	6.4	260	1220	14.7	307	1265	9.3	361	1318	3.1	9.2
Mahalanobis [28]	✗	20.9	13.1	25.8	13.5	4.1	295	1265	10.7	353	1321	4.0	385	1352	0.3	4.7
MC dropout [36]	✗	4.9	50.3	9.0	5.5	0.9	356	2322	2.3	375	2339	0.9	387	2351	0.1	1.0
Void classifier [6]	✓	10.4	41.5	23.3	6.3	5.7	350	403	9.2	365	421	5.5	381	435	1.7	5.4
Embedding density [6]	✗	0.8	46.4	2.0	35.6	1.3	145	11166	4.1	244	11271	2.4	370	11393	0.3	2.3
Image resynthesis [32]	✗	37.5	4.7	38.8	16.5	6.8	286	887	14.8	333	931	8.0	374	970	2.0	8.4
Road inpainting [31]	✗	55.8	56.8	63.2	70.5	4.8	29	5453	11.6	59	5488	10.6	171	5599	7.0	10.1
SynBoost [5]	✓	71.3	3.2	70.8	44.3	27.5	136	388	49.0	185	440	39.4	283	538	20.4	37.6
Maximized entropy [9]	✓	85.1	0.8	79.6	47.9	41.4	136	202	59.9	177	250	49.7	247	321	33.2	48.5

Table 2: Benchmark results for our RoadObstacle21 dataset. This dataset contains 388 ground truth components in total.

In our experiments, we additionally include the average sIoU per component $\overline{\text{sIoU}}$, which can be computed w.r.t. ground truth components $k \in \mathcal{K}$ or 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 (see section 3.3), we average these quantities over different thresholds $\tau \in \mathcal{T} = \{0.25, 0.30, \ldots, 0.75\}$, similar to [30], yielding the averaged component-wise F_1 score $\overline{F_1} = \frac{1}{|\mathcal{T}|} \sum_{\tau \in \mathcal{T}} F_1(\tau)$.

Discussion of the Results. Our benchmark results for RoadAnomaly21 and RoadObstacle21 are summarized in table 1 and table 2, 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 autoencoders, namely image resynthesis and SynBoost, show to be better suited for 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 discrepancy module, and they are outperformed in our experiments 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 the supplementary material. In addition, we observe a much lower component-wise $\overline{F_1}$ score than a pixel-wise one, 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 3 we summarize our results for Fishyscapes LostAndFound [6], a validation set of 100 LostAndFound images

		RoadAnomaly21		Fishyscapes LostAndFound Validation					RoadObstacle21		LostAndFound Filtered				
				Pixel-level		Component-level					Pixel-level		Component-level		
Method	OoD data	AuPRC ↑	$\overline{F_1}$ ↑	Anomaly scores AuPRC↑	FPR$_{95}$ ↓	$k \in \mathcal{K}$ $\overline{\text{sIoU}}$ ↑	$\hat{k} \in \hat{\mathcal{K}}$ $\overline{\text{sIoU}}$ ↑	$\overline{F_1}$ ↑	AuPRC ↑	$\overline{F_1}$ ↑	Anomaly scores AuPRC ↑	FPR$_{95}$ ↓	$k \in \mathcal{K}$ $\overline{\text{sIoU}}$ ↑	$\hat{k} \in \hat{\mathcal{K}}$ $\overline{\text{sIoU}}$ ↑	$\overline{F_1}$ ↑
Maximum softmax [21]	✗	27.9	5.2	5.6	40.5	3.5	2.6	1.7	15.7	6.3	30.1	33.2	14.2	14.9	10.3
ODIN [29]	✗	33.4	5.2	16.6	38.5	10.3	10.1	10.3	22.1	9.2	52.9	30.0	39.8	27.0	34.5
Mahalanobis [28]	✗	20.1	2.7	32.9	8.7	19.6	15.3	17.7	20.9	4.7	55.0	12.9	33.8	16.8	22.1
MC dropout [36]	✗	29.1	4.4	10.3	46.4	7.1	6.6	6.1	4.9	9.0	36.8	35.5	17.4	11.8	13.0
Void classifier [6]	✓	36.9	6.5	11.4	15.3	9.0	24.3	13.9	10.4	23.3	4.8	47.0	1.8	3.3	1.9
Embedding density [6]	✗	37.6	7.9	8.9	42.2	5.9	4.4	4.8	0.8	2.0	61.7	10.4	37.8	20.3	27.5
Image resynthesis [32]	✗	52.3	12.4	5.1	29.8	23.3	2.3	4.3	37.5	8.4	57.1	8.8	3.5	15.4	3.7
Road inpainting [31]	✗	-	-	-	-	-	-	-	55.8	10.1	78.5	49.9	50.7	40.8	51.3
SynBoost [5]	✓	56.6	10.2	65.0	31.0	28.1	36.3	36.8	71.3	37.6	81.7	4.6	36.8	51.6	48.7
Maximized entropy [9]	✓	85.6	29.0	44.3	37.7	21.1	31.3	28.4	85.1	48.5	77.9	9.7	45.9	43.0	49.9

Table 3: Benchmark results for Fishyscapes LostAndFound validation and LostAndFound filtered, 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 1 and table 2, are additionally included for cross evaluation (gray columns).

[40] with refined labels fitting the anomaly track, and LostAndFound itself, with original labels fitting the obstacle track. Note that for LostAndFound we filtered out all images that contain humans and bicycles labeled as obstacles 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 pixel-level AuPRC and component-level $\overline{F_1}$ scores, between the methods 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 particular, 13.8% of the image pixels in RoadAnomaly21 belong to anomalies, *vs*. only 0.23% in Fishyscapes LostAndFound. Thus, we expect that results on RoadAnomaly21 are statistically more reliable than on Fishyscapes LostAndFound. In the supplementary material we provide further and more fragmented results, in terms of both objects categories and object sizes.

Finally, we also applied our benchmark suite to the LiDAR guided Small obstacle Segmentation dataset [42]. 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 illumation), making this dataset less suitable to camera-only methods. We present the corresponding results in the supplementary material.

6. 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. They challenge two important abilities of computer vision systems, on one hand the ability to detect and localize unknown objects, on the other the 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 anomaly segmentation. The datasets and the software are available at http://www.segmentmeifyoucan.com/.

Acknowledgement

Robin Chan and Svenja Uhlemeyer acknowledge funding by the German Federal Ministry for Economic Affairs and Energy, within the projects "KI Absicherung", grant no. 19A19005R, and "KI Delta Learning", 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 in collecting and labeling of data.

References

[1] Matt Angus, Krzysztof Czarnecki, and Rick Salay. Efficacy of pixel-level OOD detection for semantic segmentation. *CoRR*, abs/1911.02897, 2019. 2, 3

[2] Andrei Atanov, Arsenii Ashukha, Dmitry Molchanov, et al. Uncertainty estimation via stochastic batch normalization. In *Advances in Neural Networks – ISNN 2019*, pages 261–269, Cham, 2019. Springer International Publishing. 3

[3] 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)*, pages 57.1–57.12. BMVA Press, September 2017. 3

[4] Petra Bevandić, Ivan Krešo, Marin Oršić, and Siniša Šegvić. Simultaneous semantic segmentation and outlier detection in presence of domain shift. In Gernot A. Fink, Simone Frintrop, and Xiaoyi Jiang, editors, *Pattern Recognition*, pages 33–47, Cham, 2019. Springer International Publishing. 3

[5] Giancarlo Di Biase, Hermann Blum, Roland Siegwart, and Cesar Cadena. Pixel-wise anomaly detection in complex driving scenes, 2021. 3, 6, 7, 8, 16, 17

[6] Hermann Blum, Paul-Edouard Sarlin, Juan Nieto, Roland Siegwart, and Cesar Cadena. Fishyscapes: A benchmark for safe semantic segmentation in autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, October 2019. 1, 2, 3, 5, 6, 7, 8, 11, 12, 13, 16, 17

[7] Kendrick Boyd, Kevin H. Eng, and C. David Page. Area under the precision-recall curve: Point estimates and confidence intervals. In Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Železný, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 451–466, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. 4

[8] Dominik Brüggemann, Robin Chan, Matthias Rottmann, Hanno Gottschalk, and Stefan Bracke. Detecting Out of Distribution Objects in Semantic Segmentation of Street Scenes. In *The 30th European Safety and Reliability Conference (ESREL)*, 2020. 2

[9] Robin Chan, Matthias Rottmann, and Hanno Gottschalk. Entropy maximization and meta classification for out-of-distribution detection in semantic segmentation, 2020. 2, 3, 6, 7, 8, 12, 16, 17

[10] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 40, pages 834–848, 2018. 12

[11] Liang Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *The European Conference on Computer Vision (ECCV)*, 9 2018. 12

[12] Hyunsun Choi, Eric Jang, and Alexander A Alemi. WAIC, but why? generative ensembles for robust anomaly detection, Oct. 2018. 2

[13] 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. 1, 2, 3, 4, 12

[14] Clement Creusot and Asim Munawar. Real-time small obstacle detection on highways using compressive rbm road reconstruction. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 162–167, 2015. 3

[15] Terrance DeVries and Graham W. Taylor. Learning Confidence for Out-of-Distribution Detection in Neural Networks, Feb 2018. 3, 11

[16] 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*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA, 20–22 Jun 2016. PMLR. 3

[17] Jakob Geyer, Yohannes Kassahun, Mentar Mahmudi, Xavier Ricou, Rupesh Durgesh, Andrew S. Chung, Lorenz Hauswald, Viet Hoang Pham, Maximilian Mühlegg, Sebastian Dorn, Tiffany Fernandez, Martin Jänicke, Sudesh Mirashi, Chiragkumar Savani, Martin Sturm, Oleksandr Vorobiov, Martin Oelker, Sebastian Garreis, and Peter Schuberth. A2D2: AEV Autonomous Driving Dataset. http://www.a2d2.audi, 2019. 1

[18] 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)*, pages 1289–1298, 2020. 3

[19] 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. 1, 2

[20] Dan Hendrycks, Steven Basart, Mantas Mazeika, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling out-of-distribution detection for real-world settings, 2020. 1, 2

[21] 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. 2, 5, 7, 8, 11, 16, 17

[22] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. *Proceedings of the International Conference on Learning Representations*, 2019. 3, 12

[23] 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)*, pages 365–370, 2017. 2

[24] 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. 2, 3, 12

[25] 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*, pages 5574–5584. Curran Associates, Inc., 2017. 3

[26] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, pages 1332–1338, 2015. 2

[27] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems 30*, pages 6402–6413. Curran Associates, Inc., 2017. 3

[28] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks. In S Bengio, H Wallach, H Larochelle, K Grauman, N Cesa-Bianchi, and R Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, pages 7167–7177. Curran Associates, Inc., 2018. 2, 7, 8, 11, 16, 17

[29] 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. 2, 5, 7, 8, 11, 16, 17

[30] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755. Springer International Publishing, 2014. 1, 6, 7, 12

[31] Krzysztof Lis, Sina Honari, Pascal Fua, and Mathieu Salzmann. Detecting road obstacles by erasing them, 2020. 3, 6, 7, 8, 16, 17

[32] Krzysztof Lis, Krishna Nakka, Pascal Fua, and Mathieu Salzmann. Detecting the unexpected via image resynthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 2, 3, 6, 7, 8, 11, 12, 16, 17

[33] David J. C. MacKay. A practical bayesian framework for backpropagation networks. *Neural Computation*, 4(3):448–472, 1992. 3

[34] A. Mehrtash, W. M. Wells, C. M. Tempany, P. Abolmaesumi, and T. Kapur. Confidence calibration and predictive uncertainty estimation for deep medical image segmentation. *IEEE Transactions on Medical Imaging*, pages 1–1, 2020. 2

[35] Alexander Meinke and Matthias Hein. Towards neural networks that provably know when they don't know. In *International Conference on Learning Representations*, 2020. 2, 3

[36] Jishnu Mukhoti and Yarin Gal. Evaluating bayesian deep learning methods for semantic segmentation, 2019. 3, 5, 7, 8, 11, 16, 17

[37] 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)*, pages 1–6, 2017. 3

[38] Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012. 3

[39] 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. 2

[40] Peter Pinggera, Sebastian Ramos, Stefan Gehrig, Uwe Franke, Carsten Rother, and Rudolf Mester. Lost and found: detecting small road hazards for self-driving vehicles. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016. 2, 8, 13, 14

[41] Matthias Rottmann, Pascal Colling, Thomas Paul Hack, Robin Chan, Fabian Hüger, Peter Schlicht, and Hanno Gottschalk. 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. 4

[42] Aasheesh Singh, Aditya Kamireddypalli, Vineet Gandhi, and K Madhava Krishna. LiDAR guided small obstacle segmentation, Mar. 2020. 2, 8, 13, 14

[43] Joost van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Simple and scalable epistemic uncertainty estimation using a single deep deterministic neural network, Mar. 2020. 2

[44] Yingda Xia, Yi Zhang, Fengze Liu, Wei Shen, and Alan Yuille. Synthesize then compare: Detecting failures and anomalies for semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 3

[45] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020. 1, 2

[46] Oliver Zendel, Katrin Honauer, Markus Murschitz, Daniel Steininger, and Gustavo Fernandez Dominguez. Wilddash-creating hazard-aware benchmarks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 402–416. openaccess.thecvf.com, 2018. 2

[47] Xiang Zhang and Yann LeCun. Universum prescription: Regularization using unlabeled data. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017. 1, 11

[48] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 12

[49] 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. 12

Supplementary Material

A. Evaluated Methods

In this section, we introduce technical details on the methods that we already evaluated on our benchmark and thus form the backbone of our leader board.

A.1. Method Description

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 $X \subseteq [0,1]^{N \times |\mathcal{Z}| \times 3}$ a dataset with N images. Below, we describe how s is obtained for each approach.

Maximum softmax probability. Let $f : \mathcal{X} \to \mathbb{R}^{|\mathcal{Z}| \times |\mathcal{C}|}$ denote the output of a semantic segmentation DNN. The maximum softmax probability (MSP) is a commonly-used baseline for OoD detection at image level [21]. It computes an anomaly score for each pixel $z \in \mathcal{Z}$ as

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

where $\sigma(\cdot) : \mathbb{R}^{|\mathcal{C}|} \to (0,1)^{|\mathcal{C}|}$ 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 [29] small perturbations are added to every pixel $z \in \mathcal{Z}$ of image x by

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

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

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

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

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

an anomaly score for each pixel z can be computed as the Mahalanobis distance [28]

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

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 samplings and $\hat{q}_m^c := \sigma(f_z^c(x))$ the softmax probability of class $c \in \mathcal{C}$ for sample $m \in \{1, \dots, M\}$. One can compute the predictive entropy as

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

As suggested in [36], the mutual information can then be used to define an anomaly score

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

Void classifier. In [15], 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 [47], and compute the anomaly score for each pixel $z \in \mathcal{Z}$ as the softmax score for the void class, which yields

$$s_z(x) = \sigma(f_z^{\text{void}}(x)), \; x \in \mathcal{X} \,. \tag{14}$$

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 [6]. 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 $u : \mathbb{R}^{|\mathcal{Z}'|} \to \mathbb{R}^{|\mathcal{Z}|}$. This yields an anomaly score for each $z \in \mathcal{Z}$ as

$$s_z(x) = u_z \left(\left(-\log \hat{p}(h_{z'}^l(x)) \right)_{z' \in \mathcal{Z}'} \right), x \in \mathcal{X} \,. \tag{15}$$

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 $g : \mathcal{C}^{|\mathcal{Z}|} \to \mathcal{X}'$ 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* [32] $d : \mathcal{C}^{|\mathcal{Z}|} \times \mathcal{X}' \times \mathcal{X} \to \mathbb{R}^{|\mathcal{Z}|}$ 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,

$$s_z(x) = d_z \left(\hat{y}(x), g(\hat{y}(x)), x \right), \; x \in \mathcal{X} \,. \tag{16}$$

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

$$s_z(x) = d_z\left(\, g'(x), x \,\right), \; x \in \mathcal{X} \,. \tag{17}$$

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

$$H_z(x) = -\sum_{c \in \mathcal{C}} \sigma(f_z^c(x)) \log\left(\sigma(f_z^c(x))\right) \tag{18}$$

and the pixel-wise softmax distance

$$D_z(x) = 1 - \max_{c \in \mathcal{C}} \sigma(f_z^c(x)) + \max_{c' \in \mathcal{C} \setminus \{\arg\max_{c \in \mathcal{C}}\}} \sigma(f_z^{c'}(x)) \tag{19}$$

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

$$s_z(x) = d_z\left(\, \hat{y}(x), g(\hat{y}(x)), x, H(x), D(x) \,\right) \,. \tag{20}$$

Maximized entropy. Starting from a pretrained DNN, a second training objective is introduced to maximize the softmax entropy on OoD pixels [9, 22, 24]. This yields the multi-criteria loss function

$$\begin{aligned} &(1-\lambda)\&\mathbb{E}_{(x,y)\sim\mathcal{D}_{in}}\left[\ell_{in}(\sigma(f_z(x)), y_z(x))\right] \\ &+ \lambda\&\mathbb{E}_{x'\sim\mathcal{D}_{out}}\left[\ell_{out}(\sigma(f_z(x')))\right] \;\; \lambda \in [0,1] \,, \end{aligned} \tag{21}$$

where ℓ_{in} is the empirical cross entropy and ℓ_{out} the negative log-likelihood 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 [30] is used whose images do not depict any object classes also available in $\mathcal{D}_{in}$, the Cityscapes dataset [13]. The COCO subset together with the Cityscapes training data are then included into a tender retraining of the Cityscapes model. The anomaly score is then computed via the softmax entropy as

$$s_z(x) = -\sum_{c \in \mathcal{C}} \sigma(f_z^c(x)) \log\left(\sigma(f_z^c(x))\right) \,. \tag{22}$$

A.2. Underlying Segmentation DNNs

Most of our evaluated methods build upon variants of DeepLab [10] 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 [11], as presented first in [6]. For maximum softmax, ODIN, Mahalanobis distance and maximized entropy, we employ a more modern DeepLabv3+ model with a WideResNet38 backbone [49]. For image resynthesis we use the more lightweight PSPNet as underlying model for semantic segmentation just like originally proposed by [32]. All these networks are initialized with publicly available weights which are pretrained on the Cityscapes dataset.

Method	Semantic segmentation Network architecture	time in s ↓ per image
Maximum softmax	DeepLabv3+ WideResNet38 backbone [49]	1.19
ODIN	DeepLabv3+ WideResNet38 backbone [49]	16.74
Mahalanobis Distance	DeepLabv3+ WideResNet38 backbone [49]	63.60
MC dropout	DeepLabv3+ Xcpection backbone [11]	19.68
Void Classifier	DeepLabv3+ Xcpection backbone [11]	2.02
Embedding density	DeepLabv3+ Xcpection backbone [11]	10.66
Image resynthesis	PSPNet [48]	1.43
Maximized entropy	DeepLabv3+ WideResNet38 backbone [49]	2.90

Table 4: Run time comparison of the different anomaly segmentation methods. The averaged inference time for one image of RoadAnomaly21 is reported in seconds.

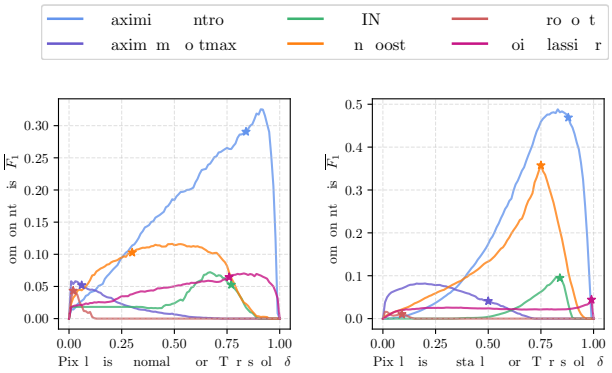
A.3. Inference Time Comparison

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 4. All methods are compared with the same hardware (NVIDIA Quadro P6000), however they might differ in the underlying network architecture.

B. Parameter Study

In our evaluation, the component-wise F_1 score (equation (5)) 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 false positive, respectively, see also equation (3) and equation (4). 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 (6)) 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.

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 perfor-



(a) RoadAnomaly21 (b) RoadObstacle21

Figure 5: The averaged component-wise metric $\overline{F_1}$ as function of the pixel-wise anomaly / obstacle threshold δ for RoadAnomaly21 and RoadObstacle21, respectively, *c.f.* table 1 and 2. The "star" marker indicates a method's $\overline{F_1}$-score at the chosen threshold δ^* according to equation (6), which is used in our default procedure for generating segmentation masks from pixel-wise anomaly / obstacle scores.

mance 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 5 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.

C. Evaluated Datasets

Besides RoadAnomaly21 and RoadObstacle21 we also performed analogous benchmark evaluations for three additional publicly available datasets: Fishyscapes LostAndFound [6], LostAndFound test set [40], and the LiDAR guided Small obstacleSegmentation dataset [42]. For the sake of comparison, we chose the Fishyscapes LostAndFound validation set for the anomaly track and the LostAndFound test set as well as the Small Obstacle dataset for the obstacle track.

C.1. Fishyscapes LostAndFound

The Fishyscapes LostAndFound validaion dataset [6] consists of 100 images from the original LostAndFound data [40] 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 7, 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.

As already discussed in section 5, 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 5. 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 1. 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 similarality of this dataet 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 10.

C.2. LostAndFound filtered

The LostAndFound dataset [40] 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 filtered. In this way, the results obtained with our evaluated methods on LostAndFound filtered and on our RoadObstacle21 dataset are comparable.

Both datasets have obstacles in the same size range. Both RoadObstacle21 and LostAndFound filtered 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 7. 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 11 and 12. Looking at the results in table 6, we observe for LostAndFound filtered, just as in Fishyscapes LostAndFound (table 5), 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 12.

C.3. LiDAR Guided Small Obstacle Segmentation Dataset

The third publicly available dataset to which we applied our benchmark suite is the LiDAR guided Small obstacle Segmentation dataset [42], which can be viewed as a reference dataset for our obstacle track. The results corresponding to this dataset are given in table 7. 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 13 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.

D. 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 8 different scenes, shown in figure 6:

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. and night images (asphalt night) .

We evaluate each subset using our benchmark suite and report the results in table 9. 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.

E. 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 8, we report the averaged sIoU (equation (3)) 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 9. 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.

F. 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 8. 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.

		Pixel-level			Component-level											
	requires	Anomaly scores			$k \in \mathcal{K}$	$\hat{k} \in \hat{\mathcal{K}}$	sIoU $> \tau = 0.25$			sIoU $> \tau = 0.50$			sIoU $> \tau = 0.75$			
Method	OoD data	AuPRC ↑	FPR_{95} ↓	F_1^* ↑	$\overline{\text{sIoU}}$ ↑	$\overline{\text{sIoU}}$ ↑	FN ↓	FP ↓	F_1 ↑	FN ↓	FP ↓	F_1 ↑	FN ↓	FP ↓	F_1 ↑	$\overline{F_1}$ ↑
Maximum softmax [21]	✗	5.6	40.5	12.1	3.5	2.6	152	209	6.7	164	221	0.5	165	222	0.0	1.7
ODIN [29]	✗	16.6	38.5	23.7	10.3	10.1	137	141	16.8	145	149	12.0	162	166	1.8	10.3
Mahalanobis [28]	✗	32.9	8.7	37.3	19.6	15.3	111	160	28.5	132	183	17.3	155	205	5.3	17.7
MC dropout [36]	✗	10.3	46.4	17.7	7.1	6.6	145	161	11.6	156	171	5.2	163	178	1.2	6.1
Void classifier [6]	✓	11.4	15.3	21.9	9.0	24.3	143	39	19.5	149	45	14.2	159	55	5.3	13.9
Embedding density [6]	✗	8.9	42.2	14.8	5.9	4.4	148	212	8.6	155	218	5.1	163	226	1.0	4.8
Image resynthesis [32]	✗	5.1	29.8	11.1	5.1	4.0	150	205	7.8	157	211	4.2	164	218	0.5	3.9
SynBoost [5]	✓	65.0	31.0	67.8	28.1	36.3	103	66	42.3	107	70	39.6	130	93	23.9	36.8
Maximized entropy [9]	✓	44.3	37.7	50.9	21.1	31.3	117	63	34.8	121	68	31.8	146	93	13.7	28.4

Table 5: Benchmark results for the Fishyscapes LostAndFound validation set. This dataset contains 165 ground truth objects.

		Pixel-level			Component-level											
	requires	Anomaly scores			$k \in \mathcal{K}$	$\hat{k} \in \hat{\mathcal{K}}$	sIoU $> \tau = 0.25$			sIoU $> \tau = 0.50$			sIoU $> \tau = 0.75$			
Method	OoD data	AuPRC ↑	FPR_{95} ↓	F_1^* ↑	$\overline{\text{sIoU}}$ ↑	$\overline{\text{sIoU}}$ ↑	FN ↓	FP ↓	F_1 ↑	FN ↓	FP ↓	F_1 ↑	FN ↓	FP ↓	F_1 ↑	$\overline{F_1}$ ↑
maximum softmax [21]	✗	30.1	33.2	32.5	14.2	14.9	1256	1222	26.8	1575	1522	8.0	1701	1647	0.5	10.3
ODIN [29]	✗	52.9	30.0	55.7	39.8	27.0	701	1552	47.2	954	1803	35.4	1319	2163	18.3	34.5
Mahalanobis [28]	✗	55.0	12.9	54.8	33.8	16.8	777	2559	35.8	1126	2911	22.4	1527	3309	7.0	22.1
MC dropout [36]	✗	36.8	35.5	42.0	17.4	11.8	1204	2072	23.6	1428	2279	13.2	1635	2484	3.5	13.0
void classifier [6]	✓	4.8	47.0	13.7	1.8	3.3	1661	864	3.7	1686	887	1.8	1704	905	0.4	1.9
embedding density [6]	✗	61.7	10.4	61.7	37.8	20.3	646	2205	42.7	963	2562	29.7	1526	3103	7.3	27.5
image resynthesis [32]	✗	57.1	8.8	55.1	27.2	15.2	947	2379	31.4	1232	2667	19.7	1560	2989	6.1	19.2
road inpainting [31]	✗	78.5	49.9	74.8	52.6	37.8	548	1222	56.7	664	1350	50.9	910	1602	38.9	49.8
SynBoost [5]	✓	81.7	4.6	75.2	36.8	51.6	775	292	63.6	942	459	52.3	1381	898	22.4	48.7
maximized entropy [9]	✓	77.9	9.7	76.8	45.9	43.0	639	777	60.2	781	911	52.3	1113	1244	33.6	49.9

Table 6: Benchmark results for the LostAndFound filtered dataset. This dataset contains 1709 ground truth objects.

		Pixel-level			Component-level											
	requires	Anomaly scores			$k \in \mathcal{K}$	$\hat{k} \in \hat{\mathcal{K}}$	sIoU $> \tau = 0.25$			sIoU $> \tau = 0.50$			sIoU $> \tau = 0.75$			
Method	OoD data	AuPRC ↑	FPR_{95} ↓	F_1^* ↑	$\overline{\text{sIoU}}$ ↑	$\overline{\text{sIoU}}$ ↑	FN ↓	FP ↓	F_1 ↑	FN ↓	FP ↓	F_1 ↑	FN ↓	FP ↓	F_1 ↑	$\overline{F_1}$ ↑
maximum softmax [21]	✗	0.7	57.1	2.2	0.5	0.4	1196	1669	0.5	1202	1675	0.1	1203	1676	0.0	0.2
ODIN [29]	✗	1.9	49.4	5.9	2.9	1.9	1148	1799	3.6	1171	1823	2.1	1196	1847	0.5	2.1
Mahalanobis [28]	✗	1.4	45.5	2.4	7.1	1.7	1039	4923	5.2	1137	5024	2.1	1198	5085	0.2	2.3
MC dropout [36]	✗	0.6	81.2	2.3	0.4	0.4	1194	1236	0.7	1201	1242	0.2	1203	1244	0.0	0.3
void classifier [6]	✓	0.8	59.6	2.1	1.5	2.1	1169	825	3.3	1193	849	1.0	1200	856	0.3	1.4
embedding density [6]	✗	0.5	66.0	1.1	9.8	0.9	1010	12535	2.8	1122	12645	1.2	1200	12721	0.0	1.3
SynBoost [5]	✓	12.4	64.8	22.9	11.2	9.8	1016	1190	14.5	1043	1216	12.4	1117	1290	6.7	11.6
maximized entropy [9]	✓	4.8	63.1	11.6	2.0	3.8	1159	606	4.7	1184	631	2.1	1202	649	0.1	2.4

Table 7: Benchmark results for the LiDAR guided Small obstacle Segmentation dataset. This dataset contains 1203 ground truth components in total.

		all anomalies			animals			vehicles			other anomalies		
	OoD	$N = 100$			$N = 59$			$N = 22$			$N = 12$		
Method	data	AuPRC ↑	FPR_{95} ↓	$\overline{F_1}$ ↑	AuPRC ↑	FPR_{95} ↓	$\overline{F_1}$ ↑	AuPRC ↑	FPR_{95} ↓	$\overline{F_1}$ ↑	AuPRC ↑	FPR_{95} ↓	$\overline{F_1}$ ↑
Maximum softmax [21]	✗	27.9	72.2	5.2	25.2	75.6	4.9	30.0	72.0	4.1	27.2	59.6	8.1
ODIN [29]	✗	33.4	72.0	5.2	32.1	72.9	4.9	29.7	74.9	4.2	38.7	63.9	9.5
Mahalanobis [28]	✗	20.1	86.9	2.7	21.3	87.4	2.5	16.7	87.4	1.9	34.9	65.4	11.7
MC dropout [36]	✗	29.1	69.9	4.4	24.8	74.0	3.2	36.3	70.3	4.5	19.7	61.6	13.6
Void classifier [6]	✓	36.9	63.4	6.5	32.2	66.9	4.0	42.6	38.7	8.7	23.3	68.7	20.5
Embedding density [6]	✗	37.6	70.8	7.9	43.9	63.2	8.4	30.1	88.4	3.5	27.3	56.6	22.8
Image resynthesis [32]	✗	52.3	25.9	12.4	51.4	26.5	16.4	52.3	25.7	5.3	43.3	53.0	13.8
SynBoost [5]	✓	56.6	62.3	10.3	54.7	66.2	10.2	57.4	62.2	7.0	47.0	63.8	21.9
Maximized entropy [9]	✓	85.6	14.9	29.0	92.2	7.2	41.9	79.0	17.9	14.8	58.0	17.6	25.4

Table 8: Effect of different of anomalies in the RoadAnomaly21 dataset. In total, RoadAnomaly21 contains 59 images with only animals, 22 images with only vehicles and 12 with other anomalies (denoted with N in the table). Images containing objects from both the animal and the vehicle category are excluded in this evaluation.

Method	OoD data	road cracked $N=40$ AuPRC	road cracked $N=40$ $\overline{F_1}$	asphalt dark $N=47$ AuPRC	asphalt dark $N=47$ $\overline{F_1}$	road gravel $N=33$ AuPRC	road gravel $N=33$ $\overline{F_1}$	asphalt gray $N=66$ AuPRC	asphalt gray $N=66$ $\overline{F_1}$	motorway $N=30$ AuPRC	motorway $N=30$ $\overline{F_1}$	sun reflection $N=72$ AuPRC	sun reflection $N=72$ $\overline{F_1}$	road bricks $N=39$ AuPRC	road bricks $N=39$ $\overline{F_1}$	asphalt night $N=30$ AuPRC	asphalt night $N=30$ $\overline{F_1}$
Maximum softmax [21]	✗	11.7	2.9	69.3	11.1	39.5	14.9	43.4	4.2	4.8	1.9	2.1	3.5	32.7	3.7	6.0	0.3
ODIN [29]	✗	15.3	4.6	75.7	14.9	68.2	40.0	76.2	7.4	10.8	8.6	3.0	5.9	51.1	6.7	8.8	4.2
Mahalanobis [28]	✗	25.9	1.4	46.7	12.3	65.9	16.8	84.7	27.7	61.2	9.1	13.9	2.0	83.6	16.1	14.2	10.2
MC dropout [36]	✗	6.6	0.8	26.3	5.8	22.6	4.9	11.1	1.9	0.4	0.0	0.2	0.1	15.2	2.1	4.8	1.3
Void classifier [6]	✓	15.8	5.6	34.6	16.7	10.0	3.0	37.8	10.3	18.6	8.2	9.4	0.5	28.2	9.0	5.8	5.2
Embedding density [6]	✗	2.5	0.8	3.3	0.8	2.8	2.2	1.8	2.3	1.1	2.8	0.1	1.1	18.3	2.5	16.7	3.2
Image resynthesis [32]	✗	47.5	8.4	40.1	10.9	77.2	40.4	66.7	18.5	24.2	11.5	34.0	7.8	12.2	2.3	24.3	4.5
Road inpainting [31]	✗	28.1	17.4	79.4	35.2	85.7	68.2	91.6	67.3	50.8	44.7	19.5	14.4	87.2	65.6	48.4	26.9
SynBoost [5]	✓	49.4	15.6	89.3	61.6	84.3	44.8	83.7	51.1	52.8	40.9	42.9	22.2	90.6	66.4	21.4	9.8
Maximized entropy [9]	✓	77.1	37.0	96.9	59.5	98.6	87.5	94.8	64.9	64.3	40.7	43.2	18.6	93.9	44.8	41.0	10.4

Table 9: Effect of different of 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 6: The scenes of our RoadObstacle21 dataset feature a variety of road surfaces.

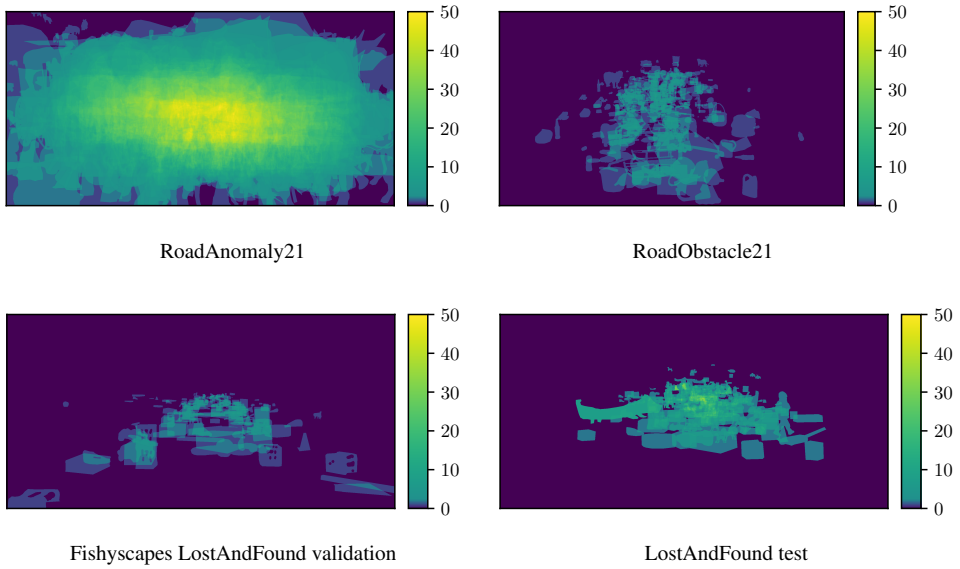
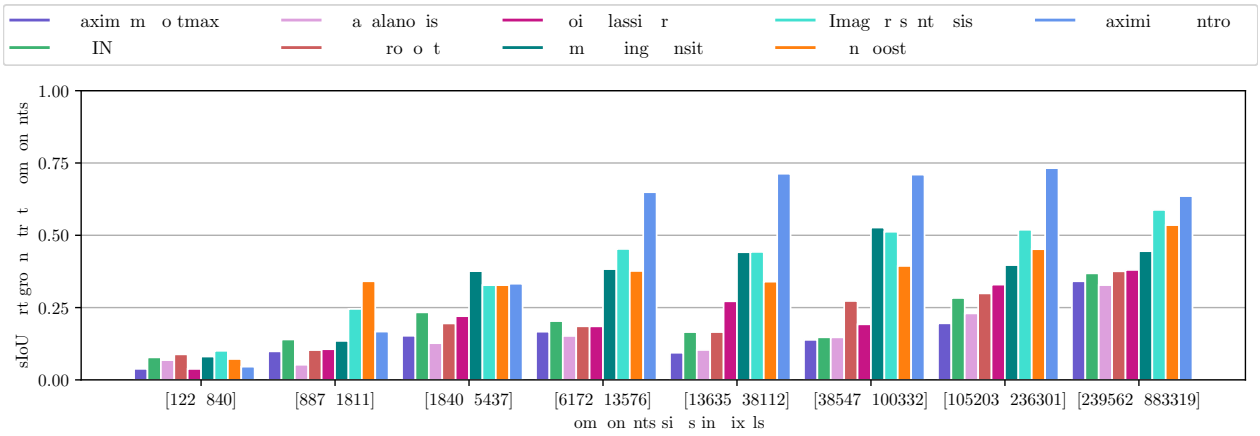
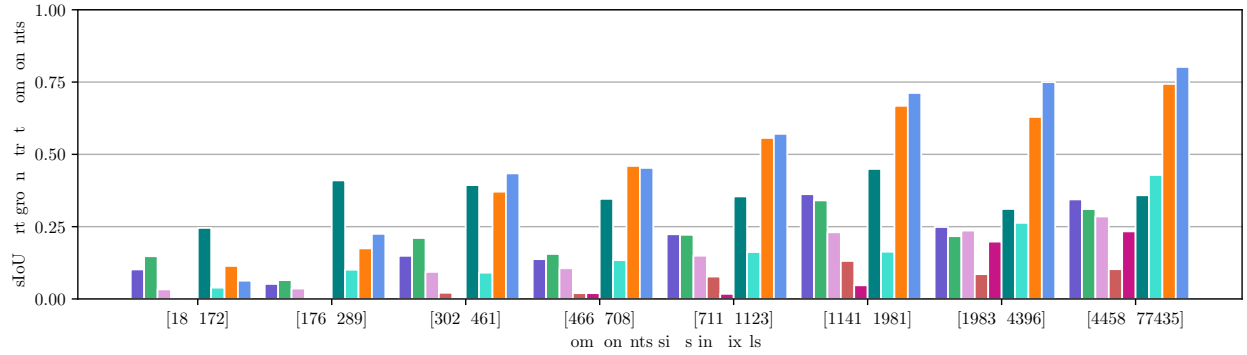


Figure 7: Comparison of the pixel distributions between RoadAnomaly21 and Fishyscapes LostAndFound (100 images each) as well as RoadObstacle21 and a subset of randomly sampled images from the LostAndFound test dataset (327 images each). The color indicates the frequency of observing an anomaly in each 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 8: 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 4 and appendix A.1.

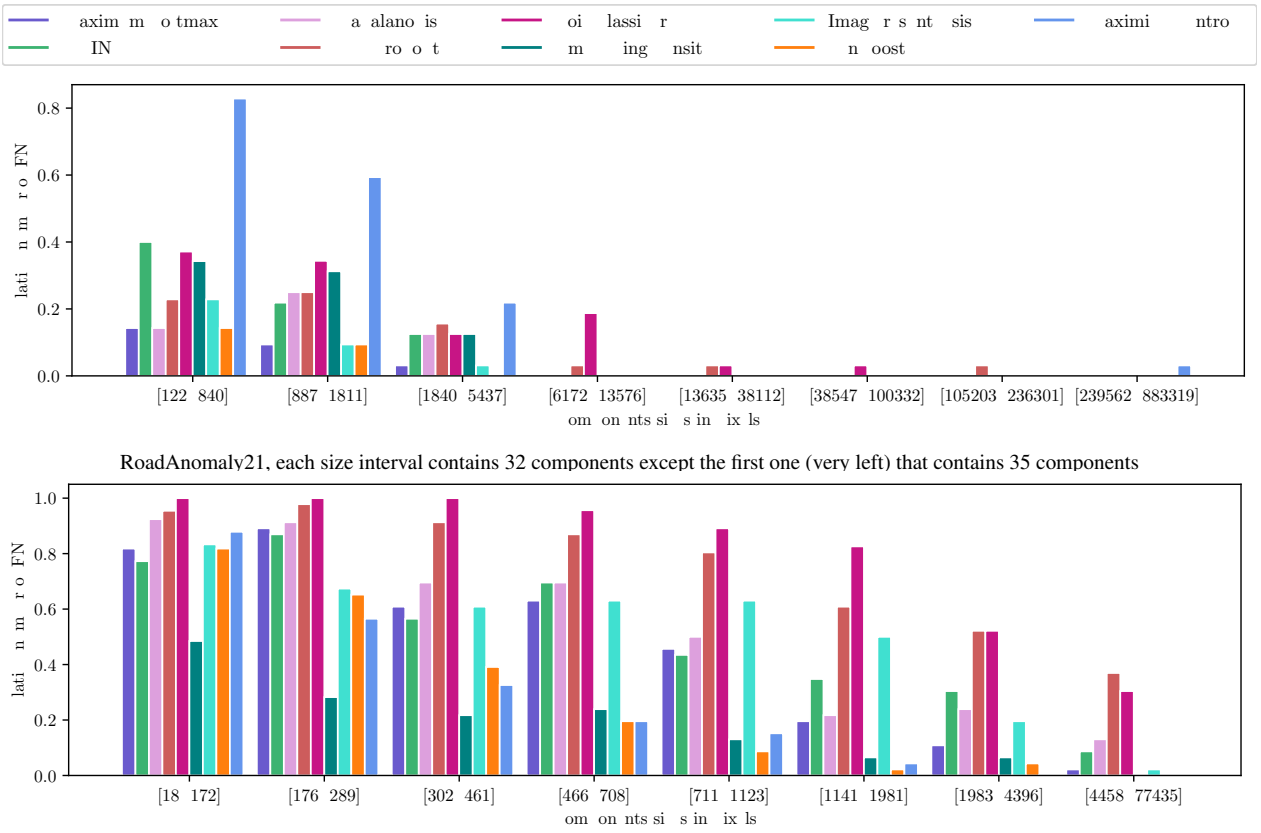


Figure 9: Comparison of the relative number of FN to TP at threshold $\tau = 0$, *i.e.* the fraction of overlooked components to the total number of ground truth components within a certain range of the components size. The evaluated methods are discussed in section 4 and appendix A.1.

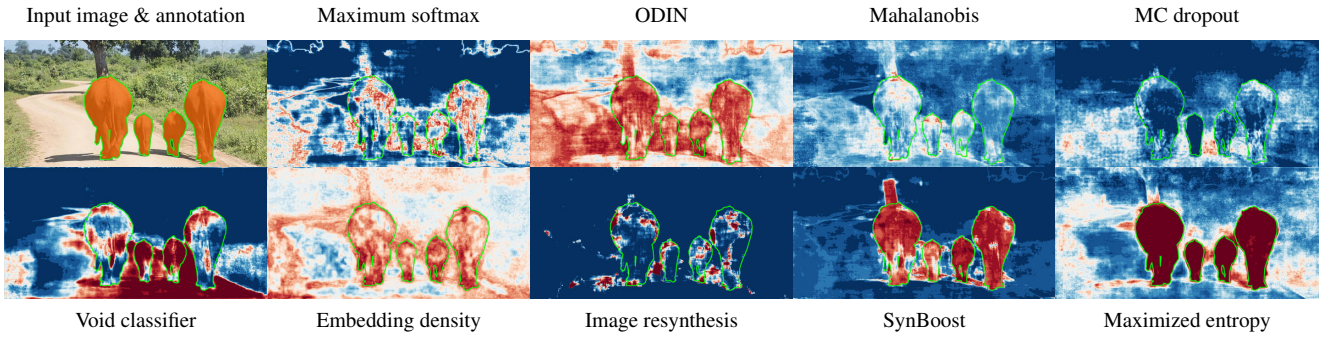


Figure 10: Qualitative comparison of the methods introduced in section 4 and appendix A.1 on a sample from RoadAnomaly21. In this example, the anomalous objects have a large size and the environment differs from scenes shown in Cityscapes. Green contours indicate the annotation of the anomaly.

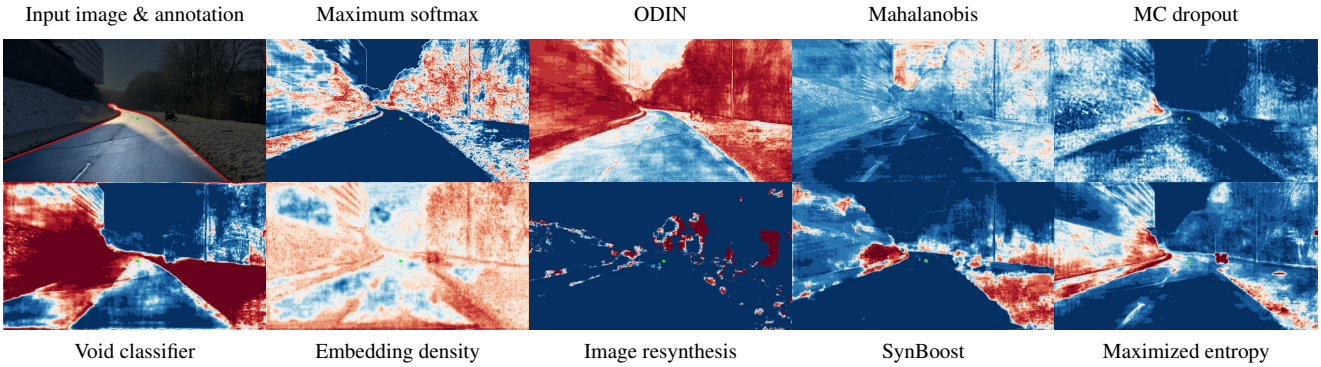


Figure 11: Qualitative comparison of the methods introduced in section 4 and appendix A.1 for an example from RoadObstacle21, where the obstacle is small and far away. Green contours indicate the annotation of the obstacle, red contours the road.

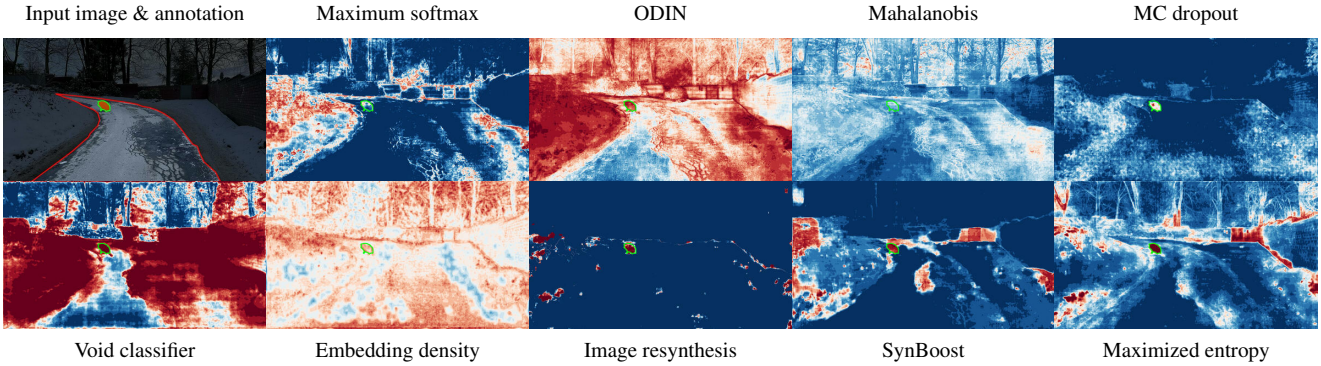


Figure 12: Qualitative comparison of the methods introduced in section 4 and appendix A.1 for an example from RoadObstacle21, showing a road surface with cracks. Green contours indicate the annotation of the obstacle, red contours the road.

(a) Input image (b) Ground truth (c) Maximized entropy anomaly score

Figure 13: 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 fail for the dataset.