SegmentMeIfYouCan: A Benchmark for Anomaly Segmentation

Robin Chan\textsuperscript{*1} Krzysztof Lis\textsuperscript{2} Svenja Uhlemeyer\textsuperscript{*1} Hermann Blum\textsuperscript{3} Sina Honari\textsuperscript{2}

Roland Siegwart\textsuperscript{3} Pascal Fua\textsuperscript{2} Mathieu Salzmann\textsuperscript{2} Matthias Rottmann\textsuperscript{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 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

\textsuperscript{*}equal contribution
\textsuperscript{1}Stochastics Group, IZMD, University of Wuppertal, Wuppertal, Germany
\textsuperscript{2}Computer Vision Laboratory, EPFL, Lausanne, Switzerland
\textsuperscript{3}Autonomous 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 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.

In this paper, motivated by the limitations of existing anomaly segmentation datasets and by the emerging body of works in this direction \cite{7910111213141516}, we introduce the SegmentMeIfYouCan \cite{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 \textit{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 download\cite{2}, 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

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 \cite{171819}, NotMNIST \cite{19}, or Omniglot \cite{20}, and separating CIFAR-10 from SVHN \cite{181921} or LSUN \cite{182122}. Such experiments can be found in many works, including \cite{171819212223}.

For semantic segmentation, a similar task was therefore proposed by the WildDash benchmark \cite{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.

\footnote{https://www.segmentmeifyoucan.com/}
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 \cite{6, 18, 21, 22, 23}. Although originally designed for image-level anomaly detection, most of these methods 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 Fishyscapes LostAndFound \cite{7} and CAOS BDD-Anomaly \cite{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 \cite{25} and the LiDAR guided Small Obstacle dataset \cite{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.

### 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 \cite{6, 18, 21, 22, 23}. Although originally designed for image-level anomaly detection, most of these methods...
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.

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 DNN’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].

### 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 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 autonomous 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 it is 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 [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 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 anomalous. On contrary, if image regions cannot be clearly assigned to any of the Cityscapes classes, they are labeled as anomalous. 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.

In RoadObstacle21, the drivable area is labeled as not obstacle. 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.

3.2 Performance Metrics

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

Pixel level. Let \( Z \) denote the set of image pixel locations. A model with a binary classifier providing anomaly scores \( s(x) \in \mathbb{R}^{\|Z\|} \) for an image \( x \in \mathcal{X} \) (from a dataset \( \mathcal{X} \subseteq [0,1]^{3 \times \|Z\| \times |N|} \) 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 (FPR95) in our evaluation. The FPR95 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.
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.

**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 [44]. In particular, while in [44] the sIoU is 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 more detail below.

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

Formally, the sIoU is a mapping \(\text{sIoU} : \hat{K} \cup \hat{K} \to [0, 1]\). For \(k \in \hat{K}\), it is defined as

\[
\text{sIoU}(k) := \frac{|k \cap \hat{K}(k)|}{|k \cup \hat{K}(k)| - |A(k)|} \quad \text{with} \quad \hat{K}(k) = \bigcup_{k \in \hat{K}, k \cap \hat{k} \neq \emptyset} \hat{k}
\]

(1)

and \(A(k) = \{z \in k' : k' \in \hat{K} \setminus \{k\}\}\). With the adjustment \(A(k)\), the pixels are excluded from the union if and only if they correctly intersect with another ground-truth component \(k' \in \hat{K}(x)\), 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 \(k \in \hat{K}\) TP if \(\text{sIoU}(k) > \tau\), and FN otherwise.

For the other error type, i.e., FP, we consider \(\hat{k} \in \hat{K}\) in Equation (1) with analogous definitions of \(\hat{K}(k)\) and \(A(\hat{k})\). We then call a predicted component \(k \in \hat{K}\) FP if \(\text{sIoU}(\hat{k}) \leq \tau\). We refer to Appendix C.2 for qualitative examples of the difference between IoU and sIoU.

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

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

(2)

which summarizes the TP, FN and FP quantities (that depend on \(\tau\)). 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.
Table 2: Benchmark results for our RoadAnomaly21 dataset. This dataset contains 388 ground-truth components in total. The main performance metrics are highlighted with gray columns.

<table>
<thead>
<tr>
<th>Method</th>
<th>Pixel-level</th>
<th>Component-level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Anomaly scores</td>
<td>Anomaly scores</td>
</tr>
<tr>
<td></td>
<td>(\text{IoU} \uparrow)</td>
<td>(\text{IoU} \uparrow)</td>
</tr>
<tr>
<td></td>
<td>(\text{FN} \downarrow)</td>
<td>(\text{FP} \downarrow)</td>
</tr>
<tr>
<td>Maximum softmax ([23])</td>
<td>✗ 28.0 72.0 34.2</td>
<td>15.5 4.8 204 778 10.6</td>
</tr>
<tr>
<td>ODIN ([22])</td>
<td>✗ 33.1 71.7 39.1</td>
<td>19.5 4.3 182 1123 10.9</td>
</tr>
<tr>
<td>Mahalanobis ([21])</td>
<td>✗ 20.0 87.0 31.9</td>
<td>14.8 2.5 206 1564 6.0</td>
</tr>
<tr>
<td>MC dropout ([22])</td>
<td>✗ 28.9 69.5 39.1</td>
<td>20.5 3.5 175 1592 9.0</td>
</tr>
<tr>
<td>Void classifier ([11])</td>
<td>✓ 36.6 63.5 44.3</td>
<td>21.1 5.5 181 987 12.2</td>
</tr>
<tr>
<td>Embedding density ([7])</td>
<td>✓ 37.5 70.8 48.7</td>
<td>33.9 5.0 107 1750 14.3</td>
</tr>
<tr>
<td>Image resynthesis ([22])</td>
<td>✓ 52.3 25.9 60.5</td>
<td>39.7 7.0 94 1422 20.1</td>
</tr>
<tr>
<td>SynBoost ([39])</td>
<td>✓ 56.4 61.9 58.0</td>
<td>34.7 6.6 111 1255 18.1</td>
</tr>
</tbody>
</table>

Maximized entropy \([11]\) | ✓ 85.5 15.0 77.4 | 49.2 17.8 85 563 35.3 | 115 586 29.5 163 626 20.1 28.7 |

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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Pixel-level</th>
<th>Component-level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Anomaly (obstacle) scores</td>
<td>Anomaly (obstacle) scores</td>
</tr>
<tr>
<td></td>
<td>(\text{IoU} \uparrow)</td>
<td>(\text{IoU} \uparrow)</td>
</tr>
<tr>
<td></td>
<td>(\text{FN} \downarrow)</td>
<td>(\text{FP} \downarrow)</td>
</tr>
<tr>
<td>Maximum softmax ([23])</td>
<td>✗ 15.7 16.6 22.5</td>
<td>19.7 4.4 255 1673 12.1</td>
</tr>
<tr>
<td>ODIN ([22])</td>
<td>✗ 22.1 15.3 30.1</td>
<td>21.4 6.4 260 1220 14.7</td>
</tr>
<tr>
<td>Mahalanobis ([21])</td>
<td>✗ 20.9 13.1 25.8</td>
<td>13.5 4.1 295 1265 10.7</td>
</tr>
<tr>
<td>MC dropout ([22])</td>
<td>✗ 4.9 50.3 9.0</td>
<td>5.5 0.9 356 2322 2.3</td>
</tr>
<tr>
<td>Void classifier ([11])</td>
<td>✓ 10.4 41.5 23.3</td>
<td>6.3 5.7 350 403 9.2</td>
</tr>
<tr>
<td>Embedding density ([7])</td>
<td>✗ 0.8 46.4 2.0</td>
<td>5.0 1.3 145 11166 4.1</td>
</tr>
<tr>
<td>Image resynthesis ([22])</td>
<td>✗ 37.7 4.7 38.8</td>
<td>16.6 6.8 286 895 14.7</td>
</tr>
<tr>
<td>Road inpainting ([42])</td>
<td>✗ 54.1 47.1 67.4</td>
<td>57.6 23.4 79 674 45.1</td>
</tr>
<tr>
<td>SynBoost ([39])</td>
<td>✓ 71.3 3.2 70.8</td>
<td>44.3 27.5 136 388 49.0</td>
</tr>
<tr>
<td>Maximized entropy ([11])</td>
<td>✓ 85.1 0.8 79.6</td>
<td>47.9 41.4 136 202 59.9</td>
</tr>
</tbody>
</table>

### 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 \([35]\);
- **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. 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 \(\delta^*\), 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, \(\delta^*\) is computed as

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

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

Moreover, for the anomaly track, components smaller than 50 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 we also allow competitors in the benchmark to submit their own anomaly segmentation masks generated via more sophisticated operations.

In our evaluation, we additionally include the average sIoU per component \(\overline{sIoU}\), which can be computed by either averaging sIoU over all ground-truth components \(k \in K\) or predicted components.
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.

Table 4: Benchmark results for Fishyscapes 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 \( \frac{1}{|T|} \), from RoadAnomaly21 and RoadObstacle21, c.f. Table 2 and Table 3, are additionally included for cross evaluation (gray columns).

**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 discrepancy module, and they are outperformed in our experiments by maximized softmax entropy, which peaks at an AuPRC of 86% and a component-wise \( F_1 \) 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
Our benchmark suite enables a unified evaluation across different datasets whenever ground truth is available. In Table 4, we summarize our results for Fishyscapes 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 $F_1$ 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”, 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 to collecting and labeling of data.

References


Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default [TODO] to [Yes], [No], or [N/A] . You are strongly encouraged to include a justification to your answer, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.
1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] The trade-off between the different tracks and their labelling strategies is described in Section 3.1. The limitations of the possible conclusions towards deployment is discussed in the Broader Impact section.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] In a particular section on Broader Impact.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]

3. If you ran experiments (e.g. for benchmarks)...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] All data and code is accessible via https://segmentmeifyoucan.com as also pointed out in the Conclusion.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We do not perform trainings in this paper, but stated the training of the tested methods in the first paragraph of section 3.3.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We took previously trained methods by authors of the models on static datasets. Therefore, the measurements are fully deterministic.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No] We do not propose models ourselves, rather take the models already given in the literature. The benchmark is public and open to submissions. Therefore, the type and amount of compute grows over time.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] All licenses are part of the official metadata record on zenodo: https://zenodo.org/communities/segmentmeifyoucan
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] All data and code is accessible via https://segmentmeifyoucan.com as also pointed out in the Conclusion.
   (d) Did you discuss whether and how consent was obtained from people whose data you’re using/curating? [Yes] We only use images that are either created by us and don’t show people or where consent is already given through the license of the image.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] The data contains no links to individuals.

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] All labels were produced by members of our labs, no external crowdsourcing was used.