Fishyscapes: A benchmark for safe semantic segmentation in autonomous driving

Hermann Blum* Paul-Edouard Sarlin* Juan Nieto Roland Siegwart Cesar Cadena
Autonomous Systems Lab, ETH Zürich

Abstract

Deep learning has enabled impressive progress in the accuracy of semantic segmentation. Yet, the ability to estimate uncertainty and detect anomalies is key for safety-critical applications like autonomous driving. Existing uncertainty estimates have mostly been evaluated on simple tasks, and it is unclear whether these methods generalize to more complex scenarios. We present Fishyscapes, the first public benchmark for uncertainty estimation in the real-world task of semantic segmentation for urban driving. It evaluates pixel-wise uncertainty estimates towards the detection of anomalous objects in front of the vehicle. We adapt state-of-the-art methods to recent semantic segmentation models and compare approaches based on softmax confidence, Bayesian learning, and embedding density. Our results show that anomaly detection is far from solved even for ordinary situations, while our benchmark allows measuring advancements beyond the state-of-the-art.

1. Introduction

Deep learning has had a high impact on the precision of computer vision methods [1–4] and enabled semantic understanding in robotic applications [5–7]. However, while these algorithms are usually compared on closed-world datasets with a fixed set of classes [8, 9], the real-world is uncontrollable, and a wrong reaction by an autonomous agent to an unexpected input can have disastrous consequences [10].

As such, to reach full autonomy while ensuring safety and reliability, decision-making systems need information about outliers and uncertain or ambiguous cases that might affect the quality of the perception output. As illustrated in Figure 1, Deep CNNs react unpredictably for inputs that deviate from their training distribution. In the presence of an outlier object (a dog), this is interpolated with available classes (road) at high confidence. Existing research to detect such behaviour is often labeled as out-of-distribution (OoD), anomaly, or novelty detection, and has so far focused on developing methods for image classification, evaluated on simple datasets like MNIST or CIFAR-10 [11–19]. How these methods generalize to more elaborate network architectures and pixel-wise uncertainty estimation has not been assessed.

Motivated by these practical needs, we introduce *Fishyscapes*[1], a benchmark that evaluates uncertainty estimates for semantic segmentation. The benchmark measures how well methods detect potentially hazardous anomalies in driving scenes. Fishyscapes is based on data from Cityscapes [9], a popular benchmark for semantic segmentation in urban driving. Our benchmark consists of (i) Fishyscapes Static, where images from Cityscapes are overlayed with objects, and (ii) Fishyscapes Lost & Found, that builds up on a road hazard dataset collected with the same setup as Cityscapes [20] and that we supplemented with labels. To further test whether methods overfit on the set of anomalous objects in these two datasets, we (iii) introduce the dynamic dataset Fishyscapes Web that updates every three months and overlays Cityscapes images with new objects found on the web.

To provide a broad overview, we adapt a variety of methods to semantic segmentation that were originally designed for image classification, with examples listed in Figure 1. Because segmentation networks are much more complex and have high computational costs, this adaptation is not trivial, and we suggest different approximations to overcome these challenges.

Our experiments show that the embeddings of intermediate layers hold important information for anomaly detection. Based on recent work on generative models, we develop a novel method using density estimation in the embedding space. However, we also show that varying visual appearance can mislead feature-based, but also other methods. None of the evaluated methods achieves the accuracy required for safety-critical applications. We conclude that these remain open problems, with our benchmark enabling the community to measure progress and build upon the best performing methods so far.

To summarize, our contributions are the following:

– The first public benchmark evaluating pixel-wise uncertainty estimates in semantic segmentation, with a dy-

*authors contributed equally
This work was supported by the Hilti Group.

[1] url ommitted for anonymous submission

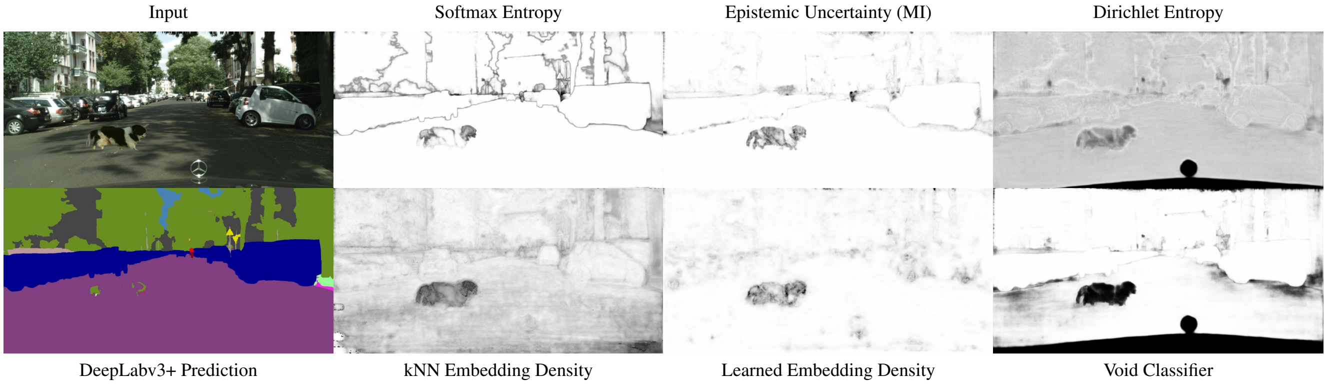

Figure 1. **Example of out-of-distribution (OoD) detection:** We evaluate the ability of Bayesian (top) and non-Bayesian (bottom) methods to segment OoD objects (here a dog) based on a semantic segmentation model. Better methods should assign a high score (dark) to pixels belonging to the object only, and a low score (white) to in-distribution (background) pixels. The semantic prediction is not sufficient.

namic, self-updating dataset for anomaly detection.

– We report an extensive evaluation with diverse state-of-the-art approaches to uncertainty estimation, adapted to the semantic segmentation task, and present a novel method for anomaly detection.

– We show a clear gap between the alleged capabilities of established methods and their performance on this real-world task, thereby confirming the necessity of our benchmark to support further research in this direction.

2. Related Work

Here we review the most relevant works in semantic segmentation and their benchmarks, and methods that aim at providing a confidence estimate of the output of deep networks.

2.1. Semantic Segmentation

State-of-the-art models are fully-convolutional deep networks trained with pixel-wise supervision. Most works [1, 21–23] adopt an encoder-decoder architecture that initially reduces the spatial resolution of the feature maps, and subsequently upsamples them with learned transposed convolution, fixed bilinear interpolation, or unpooling. Additionally, dilated convolutions or spatial pyramid pooling enlarge the receptive field and improve the accuracy.

Popular benchmarks compare methods on the segmentation of objects [24] and urban scenes. In the latter case, Cityscapes [9] is a well-established dataset depicting street scenes in European cities with dense annotations for a limited set of classes. Efforts have been made to provide datasets with increased diversity, either in terms of environments, with WildDash [25], which incorporates data from numerous parts of the world, or with Mapillary [26], which adds many more classes. Like ours, some datasets are explicitly derived from Cityscapes, the most relevant being Foggy Cityscapes [27], which overlays synthetic fog onto the original dataset to evaluate more difficult driving conditions. The Robust Vision Challenge[2] also assesses generalization of learned models across different datasets.

Robustness and reliability are only evaluated by all these benchmarks through ranking methods according to their accuracy, without taking into accounts the uncertainty of their predictions. Additionally, despite one cannot assume that models trained with closed-world data will only encounter known classes, these scenarios are rarely quantitatively evaluated. To our knowledge, WildDash [25] is the only benchmark that explicitly reports uncertainty w.r.t. OoD examples. These are however drawn from a very limited set of full-image outliers, while we introduce a diverse set of objects, as WildDash mainly focuses on accuracy.

Bevandic *et al*. [28] experiment with OoD objects for semantic segmentation by overlaying objects on Cityscapes images in a manner similar to ours. They however assume the availability of a large OoD dataset, which is not realistic in an open-world context, and thus mostly evaluate supervised methods. In contrast, we assess a wide range of methods that do not require OoD data. Mukhoti & Gal [29] introduce a new metric for uncertainty evaluation and are the first to quantitatively assess misclassification for segmentation. Yet they only compare few methods on normal in-distribution (ID) data.

2.2. Uncertainty estimation

There is a large body of work that aims at detecting OoD data or misclassification by defining uncertainty or confidence estimates.

The softmax score , *i.e*. the classification probability of the predicted class, was shown to be a first baseline [13], although sensitive to adversarial examples [30]. Its performance was improved by ODIN [31], which applies noise to

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

the input with the Fast Gradient Sign Method (FGSM) [30] and calibrates the score with temperature scaling [32].

Bayesian deep learning [33, 34] adopts a probabilistic view by designing deep models whose outputs and weights are probability distributions instead of point estimates. Uncertainties are then defined as dispersions of such distributions, and can be of several types. *Epistemic* uncertainty, or model uncertainty, corresponds to the uncertainty over the model parameters that best fit the training data for a given model architecture. As evaluating the posterior over the weights is intractable in deep non-linear networks, recent works perform Monte-Carlo (MC) sampling with dropout [35] or ensembles [36]. *Aleatoric* uncertainty, or data uncertainty, arises from the noise in the input data, such as sensor noise. Both have been applied to semantic segmentation [34], and successively evaluated for misclassification detection [29], but only on data and not for anomaly detection. Malinin & Gales [11] later suggested that *epistemic* and *aleatoric* uncertainties are only meaningful for inputs that match the training distribution, and that a third kind, *distributional* uncertainty, is required to represent model misspecification with respect to OoD inputs. Their approach however was only applied to image classifications on toy datasets, and requires OoD data during the training stage. To address the latter constraint, Lee *et al*. [37] earlier proposed a Generative Adversarial Network (GAN) that generates OoD data as boundary samples. This is however not possible for complex and high-dimensional data like high-resolution images of urban scenes.

OoD and novelty detection is often tackled by non-Bayesian approaches, which explicitly do not require examples of OoD data at training time. As such, feature introspection amounts to measuring discrepancies between distributions of deep features of training data and OoD samples, using either nearest neighbour (NN) statistics [12, 38] or Gaussian approximations [14]. These methods have the benefit of working on any classification model without requiring specific training. On the other hand, approaches specifically tailored to perform OoD detection include one-class classification [15, 16], which aim at creating discriminative embeddings, density estimation [17, 39], which estimate the likelihood of samples w.r.t to the true data distribution, and generative reconstruction [18, 19], which use the quality of auto-encoder reconstructions to discriminate OoD samples. Richter *et al*. [40] apply the latter to simple real images recorded by a robotic car and successfully detect new environments. Yet all of these methods are only applied to image classification models for OoD detection on toy datasets or for adversarial defense. As such, it is not trivial to adapt these methods to the more complex architectures used in semantic segmentation, and to the scale required by large input images.

3. Benchmark Design

In the following we describe our *Fishyscapes* benchmark: (i) the overall motivations and philosophy; (ii) the datasets and their creation; and (iii) the metrics used for comparisons of methods.

3.1. Philosophy

Because it is not possible to produce ground truth for uncertainty values, evaluating estimators is not a straightforward task. We thus compare them on the the proxy classification task [13] of detecting anomalous inputs. The uncertainty estimates are seen as scores of a binary classifier that compares the score against a threshold and whose performance reflects the suitability of the estimated uncertainty for anomaly detection. Such an approach however introduces a major issue for the design of a public OoD detection benchmark. With a publicly available ID training dataset A and OoD inputs B, it is not possible to distinguish between an uncertainty method that informs a classifier to discriminate A from any other input, and a classifier trained to discriminate A from B. The latter option clearly does not represent progress towards the goal of general uncertainty estimation, but rather overfitting.

To this end, we (i) only release a small validation set with associated ground truth masks, while keeping larger test sets hidden, and (ii) continuously evaluate submitted methods against a dynamically changing dataset. This setup preserves the uncertainty as to which anomalous objects might be encountered in the real world. To encourage unsupervised methods, we stress that the validation set is for parameter tuning only, and should not be used to train models. The evaluation is performed remotely using executables submitted by the participants.

Using these executables, methods submitted to the benchmark are continuously evaluated on every new version of the dynamic dataset. This enables us to evaluate methods on data that was not existent at the time of submission to the benchmark, assessing their generalization capabilities. Independent from their submission time, methods can always be compared using the fixed datasets.

While some of the datasets are synthetically generated, the Lost & Found data allows to check the consistency of results between real and synthetic data to identify methods that rather detect processed images than anomalies.

3.2. Datasets

FS Static is based on the validation set of Cityscapes [9]. It has a limited visual diversity, which is important to make sure that it contains none of the overlayed objects. In addition, background pixels originally belonging to the void class are excluded from the evaluation, as they may be borderline OoD. Anomalous objects are extracted from the generic Pascal VOC [24] dataset using the associated segmentation masks. We only overlay objets from classes that cannot be found in Cityscapes: *aeroplane, bird, boat, bottle, cat, chair, cow, dog, horse, sheep, sofa, tvmonitor*. Objects cropped by the image borders or objects that are too small to be seen are filtered out. We randomly size and position the objects on the underlying image, making sure that none of the objects appear on the ego-vehicle. Objects from mammal classes have a higher probability of appearing on the lower-half of the screen, while classes like birds or airplanes have a higher probability for the upper half. The placing is not further limited to ensure each pixel in the image, apart from the ego-vehicle, is comparably likely to be anomalous. To match the image characteristics of cityscapes, we employ a series of postprocessing steps similar to those described in [41], without those steps that require 3D models of the objects to e.g. adapt shadows and lighting.To make the task of anomaly detection harder, we add synthetic fog [42, 43] on the in-distribution pixels with a per-image probability. This prevents fraudulent methods to compare the input against a fixed set of Cityscapes images. The dataset is split into a minimal public validation set of 30 images and a hidden test set of 1000 images. It contains in total around 4.5e7 OoD and 1.8e9 ID pixels. The validation set only contains a small disjoint set of pascal objects to prevent few-shot learning on our data creation method.

FS Web is built similarly to FS Static, but with overlay objects crawled from the internet using a list of keywords. Our script searches for images with transparent background, uploaded in a recent timeframe, and filters out images that are too small. The only manual process is filtering out images that are not suitable, *e.g.* with decorative borders. The dataset for March 2019 contains 4.9e7 OoD and 1.8e9 ID pixels and is not publicly released. As the diversity of images and color distributions for the images from the web is much greather than those from Pascal VOC, we also adapt our overlay procedure. In total, we follow these steps, some of which were however only added for the FS Web June dataset:

- in case the image does not already have a smooth alpha channel, smooth the mask of the objects around the borders for a small transparency gradient
- adapt the brightness of the object towards the mean brightness of the overlayed pixels
- apply the inverse color histogram of the Cityscapes image to shift the color distribution towards the one found on the underlying image (FS Web Mar has a different color postprocessing)
- radial motion blur (only FS Web June)
- depth blur based on the position in the image (only FS Web June)
- color noise
- glow effects to simulate overexposure (only FS Web June)

As indicated, the postprocessing was improved between iterations of the dataset. Because the purpose of the FS Web dataset is to measure any possible overfitting of the methods through a dynamically changing dataset, we will continue to refine also this image overlay procedure at every iteration of the dataset, updating our method with recent research results.

FS Lost & Found is based on the original Lost & Found dataset [20]. However, the original dataset only included annotations for the anomalous objects and a coarse annotation of the road. It does not allow for appropriate evaluation of anomaly detection, as objects and road are very distinct in texture and it is more challenging to evaluate the anomaly score of the objects compared to building structures. In order to make use of the full image, we add pixel-wise annotations that distinguish between *objects* (the anomalies), *background* (classes contained in Cityscapes) and *void* (anything not contained in Cityscapes classes). Additionally, we filter out those sequences where the 'road hazards' are children or bikes, because these are part of regular Cityscapes data and not visual anomalies. We subsample the repetitive sequences, labelling at least every sixth image, and remove images that do not contain objects. In total, we present a public validation set of 100 images and a testset of 275 images, based on disjoint sets of locations.
While the Lost & Found images were captured with the same setup as Cityscapes, the distribution of street scenery is very different. The images were captured in small streets of housing areas, industrial areas, or on big parking lots. The anomalous objects are usually very small and are not equally distributed on the image. Nevertheless, the dataset allows to test for real images as opposed to synthetic data, therefore preventing any overfitting on synthetic image processing. This is especially important for parameter tuning on the validation set.

3.3. Metrics

We consider metrics associated with a binary classification task. Since the ID and OoD data is unbalanced, metrics based on the receiver operating curve (ROC) are not suitable. We therefore base the ranking and primary evaluation on the average precision (AP). However, as the number of false positives in high-recall areas is also relevant for safety-critical applications, we additionally report the false positive rate (FPR) at 95% true positive rate (TPR). This metric was also used in [13] and emphasizes safety.

Semantic classification is not the goal of our benchmark, but uncertainty estimation and outlier detection should not come at high cost of performance. We therefore additionally report the mean intersection over union (IoU) of the semantic segmentation on the Cityscapes validation set.

4. Evaluated Methods

We now present the methods that are evaluated in Fishyscapes. In a first part, we describe the existing baselines and how we adapted them to the task of semantic segmentation. A novel method based on learned embedding density is then presented. All approaches are applied to the state-of-the-art semantic segmentation model DeepLab-v3+ [1].

4.1. Baselines

Softmax score. The maximum softmax probability is a commonly used baseline and was evaluated in [13] for OoD detection. We apply the metric pixel-wise and additionally measure the softmax entropy, as proposed by [37], which captures more information from the softmax.

Training with OoD. While we generally strive for methods that are not biased by data, learning confidence from data is an obvious baseline and was explored in [44]. As we are not supposed to know the true known distribution, we do not use Pascal VOC, but rather approximate unknown pixels with the Cityscapes *void* class. In our evaluation, we (i) train a model to maximise the softmax entropy for OoD pixels, or (ii) introduce void as an additional output class and train with it. The uncertainty is then measured as (i) the softmax entropy, or (ii) the score of the void class.

Bayesian DeepLab was introduced by Mukhoti & Gal [29], following Kendall & Gal [34], and is the only uncertainty estimate already applied to semantic segmentation in the literature. The epistemic uncertainty is modeled by adding Dropout layers to the encoder, and approximated by T MC samples, while the aleatoric uncertainty corresponds to the spread of the categorical distribution. The total uncertainty is the predictive entropy of the distribution $\mathbf{y}$,

$$\hat{\mathbb{H}}\left[\mathbf{y}|\mathbf{x}\right] = -\sum_c \left(\frac{1}{T}\sum_t y_c^t\right)\log\left(\frac{1}{T}\sum_t y_c^t\right), \qquad (1)$$

where y_c^t is the probability of class c for sample t. The epistemic uncertainty is measured as the mutual information (MI) between $\mathbf{y}$ and the weights $\mathbf{w}$,

$$\hat{\mathbb{I}}\left[\mathbf{y},\mathbf{w}|\mathbf{x}\right] = \hat{\mathbb{H}}\left[\mathbf{y}|\mathbf{x}\right] - \frac{1}{T}\sum_{c,t} y_c^t \log y_c^t. \qquad (2)$$

Dirichlet Prior Networks [11] extend the framework of [33] by considering the predicted logits $\mathbf{z}$ as log concentration parameters $\boldsymbol{\alpha}$ of a Dirichlet distribution, which is a prior of the predictive categorical distribution $\mathbf{y}$. Intuitively, the spread of the Dirichlet prior should model the distributional uncertainty, and remain separate from the data uncertainty modelled by the spread of the categorical distribution. To this end, Malinin & Gales [11] advocate to train the network with the objective:

$$\begin{aligned}\mathcal{L}(\theta) = {} & \mathbb{E}_{p_{\text{in}}}\left[\text{KL}\left[\text{Dir}(\boldsymbol{\mu}|\boldsymbol{\alpha}_{\text{in}})||p(\boldsymbol{\mu}|\mathbf{x};\boldsymbol{\theta})\right]\right] \\ & + \mathbb{E}_{p_{\text{out}}}\left[\text{KL}\left[\text{Dir}(\boldsymbol{\mu}|\boldsymbol{\alpha}_{\text{out}})||p(\boldsymbol{\mu}|\mathbf{x};\boldsymbol{\theta})\right]\right] \\ & + \text{CrossEntropy}(\mathbf{y},\mathbf{z}).\end{aligned} \qquad (3)$$

The first term forces ID samples to produce sharp priors with a high concentration $\boldsymbol{\alpha}_{\text{in}}$, computed as the product of smoothed labels and a fixed scale α_0. The second term forces OoD samples to produce a flat prior with $\boldsymbol{\alpha}_{\text{out}} = \mathbf{1}$, effectively maximizing the Dirichlet entropy, while the last one helps the convergence of the predictive distribution to the ground truth. We model pixel-wise Dirichlet distributions, approximate OoD samples with void pixels, and measure the Dirichlet differential entropy.

kNN Embedding. Different works [12, 38] estimate uncertainty using kNN statistics between inferred embedding vectors and their neighbors in the training set. They then compare the classes of the neighbors to the prediction, where discrepancies indicate uncertainty. In more details, a given trained encoder maps a test image $\mathbf{x}'$ to an embedding $\mathbf{z}'_l = \mathbf{f}_l(\mathbf{x}')$ at layer l, and the training set $\mathbf{X}$ to a set of neighbors $\mathbf{Z}_l := \mathbf{f}_l(\mathbf{X})$. Intuitively, if $\mathbf{x}'$ is OoD, then $\mathbf{z}'$ is also differently distributed and has *e.g.* neighbors with different classes. Adapting these methods to semantic segmentation faces two issues: (i) The embedding of an intermediate layer of DeepLab is actually a map of embeddings, resulting in more than 10,000 kNN queries for each layer, which is computationally infeasible. We follow [38] and pick only one layer, selected using the FS Lost & Found validation set. (ii) The embedding map has a lower resolution than the input and a given training embedding $\mathbf{z}_l^{(i)}$ is therefore not associated with one, but with multiple output labels. As a baseline approximation, we link $\mathbf{z}_l^{(i)}$ to all classes in the associated image patch. The relative density [38] is then:

$$D(\mathbf{z}') = \frac{\sum_{i\in K, c'=c_i} \exp\left(-\frac{\mathbf{z}'\mathbf{z}^{(i)}}{|\mathbf{z}'|\,|\mathbf{z}^{(i)}|}\right)}{\sum_{i\in K} \exp\left(-\frac{\mathbf{z}'\mathbf{z}^{(i)}}{|\mathbf{z}'|\,|\mathbf{z}^{(i)}|}\right)}. \qquad (4)$$

Here, c_i is the class of $\mathbf{z}^{(i)}$ and c' is the class of $\mathbf{z}'$ in the downsampled prediction. In contrast to [38], we found that the cosine similarity from [12] works well without additional losses. Finally, we upsample the density of the feature map to the input size, assigning each pixel a density value.

As the class association in unclear for encoder-decoder architectures, we also evaluate the density estimation with k neighbors independent of the class:

$$D(\mathbf{z}') = \sum_{i\in K} \exp\left(-\frac{\mathbf{z}'\mathbf{z}^{(i)}}{|\mathbf{z}'|\,|\mathbf{z}^{(i)}|}\right). \qquad (5)$$

This assumes that an OoD sample $\mathbf{x}'$, with a low density w.r.t $\mathbf{X}$, should translate into $\mathbf{z}'$ with a low density w.r.t. $\mathbf{Z}_l$.

4.2. Learned Embedding Density

We now introduce a novel approach that takes inspiration from density estimation methods while greatly improving their scalability and flexibilty.

Density estimation using kNN has two weaknesses. First, the estimation is a very coarse isotropic approximation, while the distribution in feature space might be significantly more complex. Second, it requires to store the embeddings of the entire training set and to run a large number of NN searches, both of which are costly, especially for large input images. On the other hand, recent works [17, 39] on OoD detection leverage more complex generative models, such as normalizing flows [45–47], to directly estimate the density of the input sample $\mathbf{x}$. This is however not directly applicable to our problem, as (i) learning generative models that can capture the entire complexity of *e.g.* urban scenes is still an open problem; and (ii) the pixel-wise density required here should be conditioned on a very (ideally infinitely) large context, which is computationally intractable.

Our approach mitigates these issues by learning the density of $\mathbf{z}$. We start with a training set $\mathbf{X}$ drawn from the unknown true distribution $\mathbf{x} \sim p^*(\mathbf{x})$, and corresponding embeddings $\mathbf{Z}_l$. A normalizing flow with parameters $\boldsymbol{\theta}$ is trained to approximate $p^*(\mathbf{z}_l)$ by minimizing the negative loglikelihood (NLL) over all training embeddings in $\mathbf{Z}_l$:

$$\mathcal{L}(\mathbf{Z}_l) = -\frac{1}{|\mathbf{Z}_l|} \sum_i \log p_{\boldsymbol{\theta}}(\mathbf{z}_l^{(i)}). \tag{6}$$

The flow is composed of a bijective function $\mathbf{g}_{\boldsymbol{\theta}}$ that maps an embedding $\mathbf{z}_l$ to a latent vector $\boldsymbol{\eta}$ of identical dimensionality and with Gaussian prior $p(\boldsymbol{\eta}) = \mathcal{N}(\boldsymbol{\eta}; 0, \mathbf{I})$. Its loglikelihood is then expressed as

$$\log p_{\boldsymbol{\theta}}(\mathbf{z}_l) = \log p(\boldsymbol{\eta}) + \log \left| \det \left(\frac{d\mathbf{g}_{\boldsymbol{\theta}}}{d\mathbf{z}} \right) \right|, \tag{7}$$

and can be efficiently evaluated for some constrained $\mathbf{g}_{\boldsymbol{\theta}}$. At test time, we compute the embedding map of an input image, and estimate the NLL of each of its embeddings. In our experiments, we use the Real-NVP bijector [45], composed of a succession of affine coupling layers, batch normalizations, and random permutations.

The benefits of this method are the following: (i) A normalizing flow can learn more complex distributions than the simple kNN kernel or mixture of Gaussians used by [14], where each embedding requires a class label, which is not available here; (ii) Features follow a simpler distribution than the input images, and can thus be correctly fit with simpler flows and shorter training times; (iii) The only hyperparameters are related to the architecture and the training of the flow, and can be cross-validated with the NLL of ID data without any OoD data; (iv) The training embeddings are efficiently summarized in the weights of the generative model with a very low memory footprint.

Input preprocessing [31] can be trivially applied to our approach. Since the NLL estimator is an end-to-end network, we can compute the gradients of the average NLL w.r.t. the input image by backpropagating through the flow and the encoder.

A flow ensemble can be built by training separate density estimators over different layers of the segmentation model, similar to [14]. However, the resulting NLL estimates cannot be directly aggregated as is, because the different embedding distributions have varying dispersions and dimensions, and thus densities with very different scales. We propose to normalize the NLL $N(\mathbf{z}_l)$ of a given embedding by the average NLL of the training features for that layer:

$$\bar{N}(\mathbf{z}_l) = N(\mathbf{z}_l) - \mathcal{L}(\mathbf{Z}_l). \tag{8}$$

This is in fact a MC approximation of the differential entropy of the flow, which is intractable. In the ideal case of a multivariate Gaussian, $\bar{N}$ corresponds to the Mahalanobis distance used by [14]. We can then aggregate the normalized, resized scores over different layers. We experiment with two strategies: (i) Using the minimum detects a pixel as OoD only if it has low likelihood through all layers, thus accounting for areas in the feature space that are in-distribution but contain only few training points; (ii) Following [14], taking a weighted average , with weights given by a logistic regression fit on the FS Lost & Found validation set, captures the interaction between the layers.

5. Discussion of Results

We show in Table 1 the results of our benchmark for the aforementioned datasets and methods. Qualitative examples of the best performing methods are shown in figure 2.

Softmax Confidence. Confirming findings on simpler tasks [14], the softmax confidence is not a reliable score for anomaly detection. While training with OoD data clearly improves the softmax-based detection, it is not significantly better than Bayesian DeepLab, that does not require such data.

Visual Diversity. For most methods, there is a clear performance gap between the data from Lost & Found and the other datasets. We attribute this to two factors. First, the dataset contains a lot of images with only very small objects. This is indicated by the AP of the random classifier, which equals to the fraction of anomalous pixels. Second, as also described earlier, the qualitative examples show a lot of false positives e.g. for the void classifier where the scene is visually different to the Cityscapes data. This coin-

method	score	FS Lost & Found AP ↑	FS Lost & Found FPR_{95} ↓	FS Static AP ↑	FS Static FPR_{95} ↓	FS Web Mar 19 AP ↑	FS Web Mar 19 FPR_{95} ↓	FS Web Jun 19 AP ↑	FS Web Jun 19 FPR_{95} ↓	requires retraining	requires OoD data	Cityscapes mIoU
Random	random uncertainty	00.3	95.0	02.5	95.0	02.6	95.0	02.8	95.0	✖	✖	80.3
Softmax	max-probability	01.8	44.8	12.9	39.8	17.7	33.6	17.8	38.1	✖	✖	80.3
	entropy	02.9	44.8	15.4	39.8	23.6	33.4	23.8	37.8			
OoD training	max-entropy	01.7	30.6	27.5	23.6	33.8	21.8	43.9	20.6	✔	✔	79.0
	void classifier	10.3	**22.1**	45.0	19.4	52.9	13.3	**56.8**	**14.7**			70.4
Bayesian DeepLab	mutual information	09.8	38.5	48.7	15.5	52.1	15.9	54.7	15.3	✔	✖	73.8
Dirichlet DeepLab	prior entropy	**34.3**	47.4	31.3	84.6	27.7	93.6	43.6	78.2	✔	✔	70.5
kNN Embedding	density	03.5	30.0	44.0	20.2	50.4	13.7	36.5	33.1	✖	✖	80.3
	relative class density	00.8	-	15.8	-	20.4	-	16.1	-			
Learned Embedding Density	single-layer NLL	03.0	32.9	40.9	21.3	61.2	10.8	30.4	34.6	✖	✖	80.3
	logistic regression	04.7	24.4	57.2	**13.4**	73.2	**6.0**	40.4	26.5		✔	
	minimum NLL	04.3	47.2	**62.1**	17.4	**78.9**	9.3	41.9	47.1		✖	

Table 1. **Benchmark Results**. The gray columns mark the primary metric of the benchmark. For relative class density, 95% TPR was not reached and therefore FPR_{95} could not be evaluated.

cides also with wrong predictions of the DeepLabv3+ classifier. Nevertheless, the nature of this data shows a clear advantage of the Dirichlet DeepLab, which in the qualitative examples shows a distinction between anomalies and these 'novel' visual appearances. This supports the idea of disentangled distributional uncertainty developed in [11].

Semantic Segmentation Accuracy. The data in table 1 additionally illustrates a tradeoff between anomaly detection and segmentation performance. Methods like Bayesian DeepLab or Void Classifier are consistently among the best methods on all datasets, but need to train with special losses that reduce the segmentation accuracy by up to 10% mIoU.

Embedding based methods come without such a tradeoff, but their performance varies greatly between the different datasets, indicating that they are sensitive to visual appearance. This is for example indicated by the performance drop from FS Web March to FS Web June, where we improved the post-processing of the object overlay, but also by the performance gap between synthetic and real data. However, scores on FPR_{95} suggest that embedding based methods can be relevant for safety-critical applications, as their false positive rate is comparably low for conservative detection thresholds.

Method Variants. The comparison for training on Cityscapes void shows that a separate void class is consistently better than maximizing the softmax entropy. A comparison between different embedding methods shows that flow-based density estimation outperforms kNN based methods on all datasets, indicating that the flow can capture better the true data distribution. Our results also indicate that a combination of multiple layers is beneficial.

Challenges in Method Adaptation. The results reveal that some methods cannot be easily adapted to semantic segmentation. For example, retraining required by special losses can impair the segmentation performance, and we found that these losses (e.g. for Dirichlet DeepLab) were often unstable during training or did not converge. Other challenges rise from the complex network structures which complicate the translation of class-based embedding methods such as deep k-nearest neighbor [12] to segmentation. This is illustrated by the performance of our naïve implementation.

6. Conclusion

In this work, we introduced *Fishyscapes*, a benchmark for anomaly detection in semantic segmentation for urban driving. Comparing state-of-the-art methods on this complex task for the first time, we draw multiple conclusions:

- The softmax output from a standard classifier is a bad indicator for anomaly detection.
- Most of the better performing methods required special losses that reduce the semantic segmentation accuracy.
- Learning anomaly detection from fixed OoD data is on par with unsupervised methods for most of the datasets.
- The proposed Learned Embedding Density is a promising direction for safety-critical applications, but shows clear performance gaps.

Overall, anomaly detection is an unsolved task. To safely deploy semantic segmentation methods in autonomous cars, further research is required. As a public benchmark, *Fishyscapes* supports the evaluation of new methods on urban driving scenarios.

References

[1] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *ECCV*, 2018. DOI: 10.1007/978-3-030-01234-2_49.

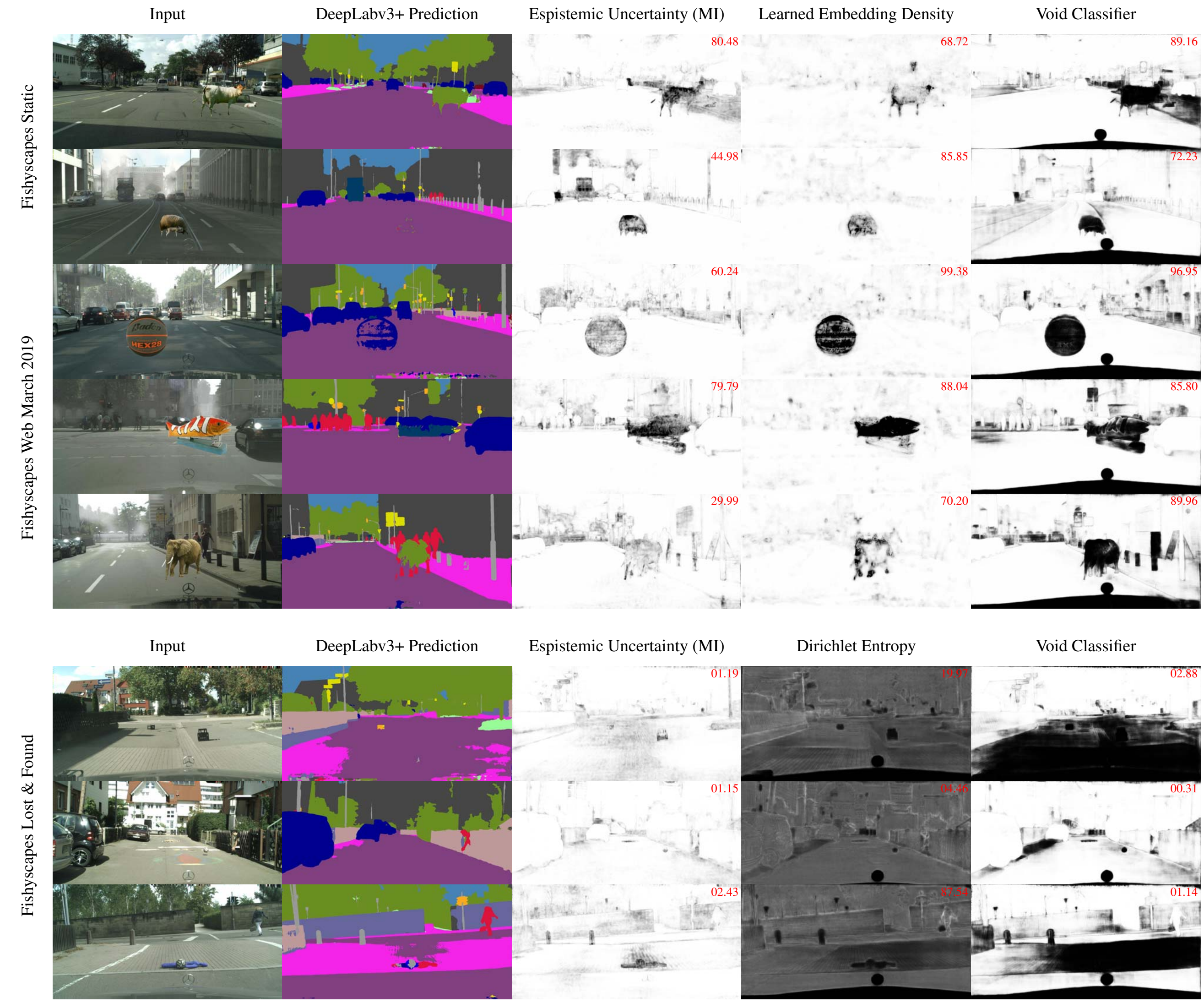

Figure 2. **Qualitative examples** in Fishyscapes Static (rows 1-2) and Fishyscapes Web March (rows 3-5) and Fishyscapes Lost & Found (rows 6-8). The best three methods per dataset are shown. Better methods should assign a high score (dark) to the overlayed object. None of the methods perfectly detects all objects. We report the AP of each score map in its top right corner.

[2] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *ICCV*, 2017. DOI: 10.1109/ICCV.2017.322.

[3] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *CVPR*, 2018. DOI: 10.1109/CVPR.2018.00214.

[4] D. Sun, X. Yang, M. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *CVPR*, 2018. DOI: 10.1109/CVPR.2018.00931.

[5] J. Mccormac, R. Clark, M. Bloesch, A. Davison, and S. Leutenegger, "Fusion++: Volumetric Object-Level SLAM," in *International Conference on 3D Vision (3DV)*, 2018. DOI: 10.1109/3DV.2018.00015.

[6] P. R. Florence, L. Manuelli, and R. Tedrake, "Dense object nets: Learning dense visual object descriptors by and for robotic manipulation," in *Conference on Robot Learning (CoRL)*, 2018.

[7] M. Liang, B. Yang, S. Wang, and R. Urtasun, "Deep continuous fusion for multi-sensor 3D object detection," in *ECCV*, 2018. DOI: 10.1007/978-3-030-01270-0_39.

[8] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *CVPR*, 2012. DOI: 10.1109/CVPR.2012.6248074.

[9] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016. DOI: 10.1109/CVPR.2016.350.

[10] D. Bozhinoski, D. Di Ruscio, I. Malavolta, P. Pelliccione, and I. Crnkovic, "Safety for mobile robotic systems: A systematic mapping study from a software engineering perspective," *J. Syst. Softw.*, vol. 151, 2019. DOI: 10.1016/j.jss.2019.02.021.

[11] A. Malinin and M. Gales, "Predictive uncertainty estimation via prior networks," in *NeurIPS*, 2018.

[12] N. Papernot and P. McDaniel, "Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning," 2018. arXiv: 1803.04765 [cs.LG].

[13] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and Out-of-Distribution examples in neural networks," in *ICLR*, 2017.

[14] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting Out-of-Distribution samples and adversarial attacks," in *NeurIPS*, 2018.

[15] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *ICML*, 2018.

[16] I. Golan and R. El-Yaniv, "Deep anomaly detection using geometric transformations," in *NeurIPS*, 2018.

[17] H. Choi, E. Jang, and A. A. Alemi, "WAIC, but why? Generative ensembles for robust anomaly detection," 2018. arXiv: 1810.01392 [stat.ML].

[18] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli, "Adversarially learned one-class classifier for novelty detection," in *CVPR*, 2018. DOI: 10.1109/cvpr.2018.00356.

[19] S. Pidhorskyi, R. Almohsen, and G. Doretto, "Generative probabilistic novelty detection with adversarial autoencoders," in *NeurIPS*, 2018.

[20] P. Pinggera, S. Ramos, S. Gehrig, U. Franke, C. Rother, and R. 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. DOI: 10.1109/IROS.2016.7759186.

[21] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015. DOI: 10.1007/978-3-319-24574-4_28.

[22] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional Encoder-Decoder architecture for image segmentation," en, *IEEE TPAMI*, vol. 39, no. 12, 2017. DOI: 10.1109/TPAMI.2016.2644615.

[23] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab," *IEEE TPAMI*, 2016. DOI: 10.1109/TPAMI.2017.2699184.

[24] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, 2010. DOI: 10.1007/s11263-009-0275-4.

[25] O. Zendel, K. Honauer, M. Murschitz, D. Steininger, and G. Fernandez Dominguez, "Wilddash-creating hazard-aware benchmarks," in *ECCV*, 2018. DOI: 10.1007/978-3-030-01231-1_25.

[26] G. Neuhold, T. Ollmann, S. R. Bulo, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *ICCV*, 2017. DOI: 10.1109/ICCV.2017.534.

[27] C. Sakaridis, D. Dai, and L. Van Gool, "Semantic foggy scene understanding with synthetic data," *International Journal of Computer Vision*, vol. 126, no. 9, 2018. DOI: 10.1007/s11263-018-1072-8.

[28] P. Bevandić, I. Krešo, M. Oršić, and S. Šegvić, "Discriminative out-of-distribution detection for semantic segmentation," 2018. arXiv: 1808.07703 [cs.CV].

[29] J. Mukhoti and Y. Gal, "Evaluating bayesian deep learning methods for semantic segmentation," 2018. arXiv: 1811.12709 [cs.CV].

[30] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *ICLR*, 2015.

[31] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," in *ICLR*, 2018.

[32] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *ICML*, 2017.

[33] Y. Gal, "Uncertainty in deep learning," PhD thesis, University of Cambridge, 2016.

[34] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" In *NIPS*, 2017.

[35] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," en, in *ICML*, 2016.

[36] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *NIPS*, 2017.

[37] K. Lee, H. Lee, K. Lee, and J. Shin, "Training confidence-calibrated classifiers for detecting Out-of-Distribution samples," in *ICLR*, 2018.

[38] A. Mandelbaum and D. Weinshall, "Distance-based confidence score for neural network classifiers," 2017. arXiv: 1709.09844 [cs.AI].

[39] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan, "Do deep generative models know what they don't know?" In *ICLR*, 2019.

[40] C. Richter and N. Roy, "Safe visual navigation via deep learning and novelty detection," in *Robotics: Science and Systems (RSS)*, 2017. DOI: 10.15607/RSS.2017.XIII.064.

[41] H. Abu Alhaija, S. K. Mustikovela, L. Mescheder, A. Geiger, and C. Rother, "Augmented reality meets computer vision: Efficient data generation for urban driving scenes," *International journal of computer vision*, vol. 126, no. 9, 2018. DOI: 10.1007/s11263-018-1070-x.

[42] C. Sakaridis, D. Dai, S. Hecker, and L. Van Gool, "Model adaptation with synthetic and real data for semantic dense foggy scene understanding," in *ECCV*, 2018. DOI: 10.1007/978-3-030-01261-8_42.

[43] D. Dai, C. Sakaridis, S. Hecker, and L. Van Gool, "Curriculum model adaptation with synthetic and real data for semantic foggy scene understanding," 2019. arXiv: 1901.01415 [cs.CV].

[44] T. DeVries and G. W. Taylor, "Learning confidence for Out-of-Distribution detection in neural networks," 2018. arXiv: 1802.04865 [stat.ML].

[45] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real NVP," in *ICLR*, 2017.

[46] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," in *NeurIPS*, 2018.

[47] L. Dinh, D. Krueger, and Y. Bengio, "NICE: Non-linear independent components estimation," 2014. arXiv: 1410.8516 [cs.LG].